



João Pedro Ribeiro da Silva Baptista

Licenciado em Ciências da Engenharia Electrotécnica e de Computadores

**Aplicação de Algoritmos Genéticos na
Localização de Dispositivos de Manobras para
Minimização de Perdas em Redes de Energia
Eléctrica**

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Francisco Alexandre Ganho da Silva Reis, Professor, Universidade Nova de Lisboa

Júri

Presidente: Professor Doutor Luís Filipe Lourenço Bernardo
Arguente: Professor Doutor João Francisco Alves Martins
Vogal: Professor Doutor Francisco Alexandre Ganho da Silva Reis



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2020

Aplicação de Algoritmos Genéticos na Localização de Dispositivos de Manobras para Minimização de Perdas em Redes de Energia Eléctrica

Copyright © João Pedro Ribeiro da Silva Baptista, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Devo, em primeiro lugar, um especial agradecimento à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa pela formação e conhecimento transmitido ao longo desta etapa.

Em seguida, quero agradecer ao Professor Doutor Francisco Reis, por toda a sua orientação e conselhos no decorrer da elaboração desta dissertação.

À minha família, por todo o apoio e incentivo que me proporcionaram, com particular destaque ao meu primo por toda a sua ajuda e disponibilidade.

Aos meus amigos, pela amizade e apoio que levarei para sempre comigo.

Por último, deixar um agradecimento aos colegas e professores que, directa ou indirectamente, me apoiaram ao longo de todo o meu percurso académico.

RESUMO

A presente dissertação endereça o problema da minimização de perdas mediante a localização óptima de dispositivos telecomandados que possibilitam, através de reconfiguração topológica, uma potencial redução de perdas numa rede de distribuição de energia eléctrica.

Foram desenvolvidas duas metodologias para o efeito: (i) baseada numa busca sequencial de manobras; (ii) mediante a aplicação de algoritmos genéticos adaptados ao problema. A metodologia baseada numa busca sequencial permite obter apenas uma solução optimizada para um determinado cenário. Aplicando os Algoritmos Genéticos estes permitem obter um conjunto de soluções de elevada qualidade que correspondem a várias topologias, todas elas diferentes entre si mas com valores de perdas semelhantes. A localização dos Órgãos de Corte de Rede, foi feita tendo em consideração todas as soluções propostas pelo algoritmo genético adaptadas a todos os cenários de carga definidos. Estas metodologias foram aplicadas a uma rede de 33 barramentos e que permitiu evidenciar a robustez e qualidade das soluções propostas sendo que, no caso desta rede, com a colocação de 8 OCR's (possibilidade de reconfigurar a rede em 8 pontos diferentes), obteve-se um potencial de redução de perdas entre 35% e 38%. Este resultado é muito relevante uma vez que as perdas em redes de distribuição tipicamente representam 8% a 12% do total de energia eléctrica distribuída.

Palavras-chave: Reconfiguração topológica, Órgãos de Corte de Rede (OCR), Telecomando, Redes de Distribuição de Energia Eléctrica, Algoritmo Genético.

ABSTRACT

The present dissertation addresses the problem of minimizing losses through the optimal location of remote-controlled devices that enable, through topological reconfiguration, a potential loss reduction in an electricity distribution network.

Two methodologies were developed for this purpose: (i) based on a sequential search for maneuvers; (ii) by applying genetic algorithms adapted to the problem. The methodology based on a sequential search allows to obtain only an optimized solution for a given scenario. Applying the Genetic Algorithms, these allow to obtain a set of high quality solutions that correspond to several topologies, all different from each other but with similar loss values. The location of the Remote-Controlled Devices was chosen taking into account all the solutions proposed by the genetic algorithm adapted to all the defined load scenarios. These methodologies were applied to a network of 33 busses which allowed to demonstrate the robustness and quality of the proposed solutions and, in the case of this network, with the placement of 8 OCR's (possibility to reconfigure the network at 8 different points), a potential loss reduction between 35% and 38%. This result is quite significant since the losses in distribution networks typically represent 8% to 12% of the total electricity distributed.

Keywords: Topological reconfiguration, Network Reclosers, Remote Control, Electricity Distribution Networks, Genetic Algorithm.

ÍNDICE

Lista de Figuras	xiii
Lista de Tabelas	xv
Siglas	xvii
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objectivos	2
1.4 Estrutura	3
2 Perdas em redes de distribuição	5
2.1 Introdução	5
2.2 Modelização do problema de perdas	5
2.3 Telecomandos em redes de distribuição	6
2.4 Avaliação de perdas	10
3 Métodos de reconfiguração topológica de redes para redução de perdas	15
3.1 Definição da metodologia	15
3.2 Método de busca sequencial	16
3.3 Algoritmo Genético	22
3.3.1 Codificação e Inicialização da população	25
3.3.2 Cruzamento	27
3.3.3 Mutação	29
3.3.4 Processo de selecção	34
3.3.5 Parametrização	35
3.4 Ambiente de desenvolvimento	36
4 Aplicação	37
4.1 Introdução	37
4.2 Caracterização da rede	37
4.3 Aplicação das diferentes metodologias à rede apresentada	38
4.3.1 Cenário: Base	38

4.3.1.1	Resultados Método de busca sequencial	39
4.3.1.2	Resultados Algoritmo Genético	41
4.3.2	Cenário: Redução de carga	45
4.3.2.1	Resultados Método de busca sequencial	46
4.3.2.2	Resultados Algoritmo Genético	47
4.3.3	Cenário: Aumento de carga	50
4.3.3.1	Resultados Método de busca sequencial	51
4.3.3.2	Resultados Algoritmo Genético	52
4.4	Proposta de localização preferencial de OCR's	56
5	Conclusões	61
5.1	Objectivos alcançados e conclusões	61
5.2	Trabalho futuro	62
	Bibliografia	65
I	Código desenvolvido	67
I.1	Funções essenciais do método de busca sequencial	67
I.2	Funções essenciais do Algoritmo Genético	71
II	Rede teste de 8 barramentos	79
II.1	Características técnicas aplicadas à rede teste	79
II.2	Resultado método de busca sequencial - Rede teste	81
III	Rede de 33 barramentos	83
III.1	Características das linhas da rede de 33 barramentos	83
III.2	Cenários de Carga aplicados à rede de 33 barramentos	85
IV	Condutores linhas aéreas CABELTE	87
V	Resultados método de busca sequencial - Rede de 33 barramentos	89
V.1	Cenário: Base	89
V.2	Cenário: Redução de carga	91
V.3	Cenário: Aumento de carga	93
VI	Resultados algoritmo genético - Rede de 33 barramentos	95
VI.1	Cenário: Base	95
VI.2	Cenário: Redução de carga	100
VI.3	Cenário: Aumento de carga	105

LISTA DE FIGURAS

1.1	Taxa de perdas em redes de Distribuição e Transporte, adaptado de [1, 2] . . .	2
2.1	Exemplo de um Interruptor Aéreo Telecomandado [14]	7
2.2	Interruptor Auto-Religador em apoio de MT	8
2.3	Órgão de Corte de Rede do Tipo 1 em apoio de MT	8
2.4	Órgão de Corte de Rede do Tipo 2 em apoio de MT	9
2.5	Órgão de Corte de Rede do Tipo 3 em apoio de MT	10
2.6	Sistema de 3 barramentos	11
3.1	Fluxograma do método de busca sequencial em <i>Python</i>	17
3.2	Rede ilustrativa - Método de busca sequencial	19
3.3	Rede ilustrativa - Método de busca sequencial (Iteração nº 1)	19
3.4	Rede ilustrativa - Método de busca sequencial (Iteração nº 2)	20
3.5	Rede ilustrativa - Método de busca sequencial (Configuração final)	21
3.6	Esquema ilustrativo dos elementos de um algoritmo genético	22
3.7	Fluxograma do algoritmo genético desenvolvido em <i>Python</i>	24
3.8	Rede ilustrativa - Algoritmo Genético	25
3.9	Codificação da rede da Figura 3.8	26
3.10	Exemplo de configuração para um vector que representa as linhas com SA . . .	26
3.11	Representação do funcionamento do <i>crossover</i>	27
3.12	<i>Crossover</i> - Configuração 1º membro	28
3.13	<i>Crossover</i> - Configuração 2º membro	29
3.14	<i>Crossover</i> - Configuração final	29
3.15	Exemplo de mutação de um indivíduo	30
3.16	Mutação simples - Rede exemplo	30
3.17	Fluxograma Mutação Inteligente	31
3.18	Mutação inteligente - Configuração inicial	32
3.19	Mutação inteligente - 1ª configuração	32
3.20	Mutação inteligente - 2ª configuração	33
3.21	Mutação inteligente - 3ª configuração	33
3.22	Mutação inteligente - 4ª configuração	34
3.23	Esquema 3D que ilustra a mutação inteligente	34

4.1	Rede <i>IEEE</i> de 33 barramentos	38
4.2	Configuração inicial da rede - Cenário: Base	38
4.3	Configuração final da rede - Cenário: Base	39
4.4	Gráfico comparativo das tensões nodais - Cenário: Base	40
4.5	Configuração final na dissertação [17] - Cenário: Base	40
4.6	Comparação das tensões nodais entre a configuração apresentada e a dissertação [17] - Cenário: Base	41
4.7	Evolução da função <i>fitness</i> ao longo das várias gerações - Cenário: Base	42
4.8	1ª configuração final (AG) - Cenário: Base	42
4.9	2ª configuração final (AG) - Cenário: Base	43
4.10	3ª configuração final (AG) - Cenário: Base	43
4.11	Gráfico comparativo das tensões nodais (AG) - Cenário: Base	44
4.12	Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Base	45
4.13	Configuração inicial da rede - Cenário: Redução	45
4.14	Configuração final da rede - Cenário: Redução de carga	46
4.15	Gráfico comparativo das tensões nodais - Cenário: Redução de carga	47
4.16	Evolução da função <i>fitness</i> ao longo das gerações - Cenário: Redução de carga	47
4.17	1ª configuração final (AG) - Cenário: Redução de carga	48
4.18	2ª configuração final (AG) - Cenário: Redução de Carga	48
4.19	3ª configuração final (AG) - Cenário: Redução de Carga	49
4.20	Gráfico comparativo das tensões nodais (AG) - Cenário: Redução de carga	49
4.21	Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Redução de carga	50
4.22	Configuração inicial da rede - Cenário: Aumento de carga	51
4.23	Configuração final da rede - Cenário: Aumento de carga	51
4.24	Gráfico comparativo das tensões nodais - Cenário: Aumento de carga	52
4.25	Evolução da função <i>fitness</i> ao longo das gerações - Cenário: Aumento de carga	52
4.26	1ª configuração final (AG) - Cenário: Aumento de Carga	53
4.27	2ª configuração final (AG) - Cenário: Aumento de Carga	53
4.28	3ª configuração final (AG) - Cenário: Aumento de Carga	54
4.29	Gráfico comparativo das tensões nodais (AG) - Cenário: Aumento de carga	54
4.30	Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Aumento de carga	55
4.31	<i>Heatmap</i> que contém o número de manobras verificadas nos 3 cenários	56
4.32	<i>Heatmap</i> que contém o número de manobras verificadas nos Cenário: Base	57
4.33	<i>Heatmap</i> que contém o número de manobras verificadas nos Cenário: Redução de carga	57
4.34	<i>Heatmap</i> que contém o número de manobras verificadas nos Cenário: Aumento de carga	58
4.35	Configuração final e localização dos respectivos OCR's	58

LISTA DE TABELAS

4.1	Tabela com os melhores valores de <i>fitness</i> - Cenário: Base	44
4.2	Tabela com os melhores valores de <i>fitness</i> - Cenário: Redução de carga	50
4.3	Tabela com os melhores valores de <i>fitness</i> - Cenário: Aumento de carga	55
4.4	Tabela com a localização e estado preferencial dos OCR's	59

SIGLAS

AG	Algoritmo Genético.
AT	Alta tensão.
END	Energia não distribuída.
ERSE	Entidade Reguladora dos Serviços Energéticos.
MT	Média tensão.
OCR	Órgãos de Corte de Rede.
OCR1	Órgãos de Corte de Rede - Tipo 1.
OCR2	Órgãos de Corte de Rede - Tipo 2.
OCR3	Órgãos de Corte de Rede - Tipo 3.
ORD	Operadores das Redes de Distribuição.
ORE	Operadores da Rede Eléctrica.

INTRODUÇÃO

1.1 Enquadramento

O aumento dos custos energéticos e a implementação de acções ambientais com o objectivo de proteger os recursos naturais, impelem as companhias de produção e distribuição de energia eléctrica a tomar medidas no sentido da redução de perdas e do aumento da eficiência energética. Por consequência, de forma a promover a redução de custos da produção e transporte de energia eléctrica é essencial, entre outras medidas, reduzir as perdas activas em redes de distribuição [12, 20].

A actividade de distribuição de energia eléctrica em Portugal está entregue à ORD e é regulada pela ERSE, sendo os exploradores da rede encorajados a operar a rede eléctrica no maior estado de optimização possível [3]. Por sua vez, a necessidade de estimação de perdas de energia e de potência eléctricas deve-se às razões seguintes: optimização e desenvolvimento de uma futura expansão da rede de distribuição; escolha da localização e da potência óptima da geração distribuída e dos equipamentos de compensação; reconfiguração dinâmica da rede, entre outros [5].

Assim, também do ponto de vista socioeconómico os ORE têm todo o interesse em investir recursos na redução de perdas, de modo a obterem um sistema optimizado [13], com custos de operação significativamente reduzidos.

1.2 Motivação

Na figura 1.1, mostra-se que as perdas de energia eléctrica na rede de transporte portuguesa se mantiveram estáveis no período 1997 a 2015, com uma média de 1.5 %.

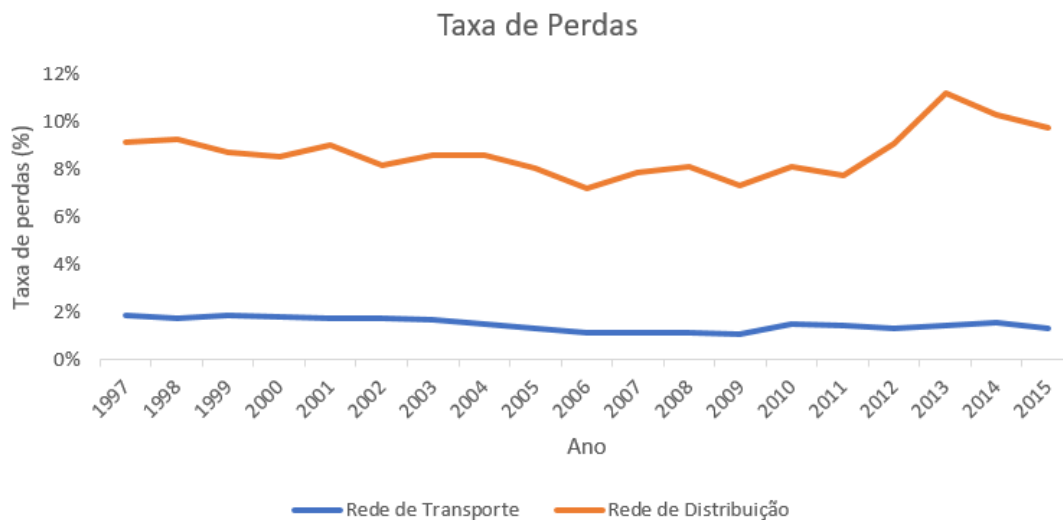


Figura 1.1: Taxa de perdas em redes de Distribuição e Transporte, adaptado de [1, 2]

Num sistema de distribuição de energia eléctrica típico, as perdas representam entre 8 % a 12 % do total da energia distribuída, o que resulta numa perda de milhões de euros por ano [15]. Verifica-se, também, que na rede de distribuição se têm registado perdas superiores, sobretudo no período de 2011 a 2014, onde se observa um pico de valor de 11.2 % face à média de 8.7 % ao longo do período registado [1, 2].

Tendo estes factores em consideração, os ORE têm vindo a procurar soluções cada vez mais fiáveis, melhorando a qualidade de serviço e optimizando o seu negócio. Num ambiente competitivo regulamentado, a qualidade da estimação de perdas é crucial para a competição nos mercados de electricidade. [5].

A análise financeira de um investimento é essencial para avaliar a rentabilidade do investimento. Deverá ser realizado um estudo benefício-custo, onde serão avaliados diversos factores: custos associados às interrupções, custos variáveis de produção e de distribuição, variação da END inicial e final, variação entre a ponta máxima de geração inicial e final, variação de perdas, taxa anual de evolução das cargas, taxa de actualização, tempo de vida útil dos equipamentos, investimento em aparelhos de corte, despesas de operação e manutenção [17]. Foram realizados diversos estudos económicos [3, 5, 15] no âmbito da redução de perdas eléctricas que fundamentam este procedimento.

1.3 Objectivos

O objectivo da presente dissertação é encontrar a localização preferencial de OCR's que possibilitem, através da reconfiguração topológica da rede de energia eléctrica, uma potencial redução de perdas, garantido a segurança da mesma.

Assim, pretende-se cumprir os objectivos seguintes:

- Desenvolver um algoritmo que através de reconfiguração topológica de uma rede seja capaz de reduzir as perdas activas dessa mesma rede;

- Aplicar Algoritmos Genéticos ao problema no sentido de procurar soluções que embora sejam topologicamente diferentes, tenham em comum valores reduzidos de perdas;
- Aplicar os algoritmos desenvolvidos a uma rede de 33 barramentos;
- Concluir sobre qual a localização preferencial de OCR's.

1.4 Estrutura

A dissertação está organizada em 5 capítulos:

- **Capítulo 1**

Neste capítulo é feito o enquadramento do tema e são definidos os objectivos principais.

- **Capítulo 2**

É explicado o problema de minimização de perdas de distribuição, são apresentados os telecomandos utilizados numa rede de distribuição de média tensão e é indicada a forma como as perdas são avaliadas para um determinado cenário topológico e de geração/carga.

- **Capítulo 3**

Tendo por base alguma da literatura consultada são definidas e propostas duas metodologias para a resolução do problema de minimização de perdas em redes de distribuição: uma tendo por base um método de busca sequencial e outra baseada em algoritmos genéticos.

- **Capítulo 4**

Aplicação de ambas as metodologias a uma rede de 33 barramentos e respectivas conclusões quanto à localização preferencial de OCR's do ponto de vista técnico.

- **Capítulo 5**

Análise dos objectivos alcançados com o trabalho desenvolvido e sugestão de perspectivas futuras a retirar do presente trabalho.

PERDAS EM REDES DE DISTRIBUIÇÃO

2.1 Introdução

O aumento dos custos de energia e as acções ambientais com objectivo de proteger os recursos naturais do planeta estimulam a que os operadores de redes de energia eléctrica promovam acções com vista à redução do uso de energia. A diminuição das perdas de energia eléctrica em redes de distribuição e transporte promove esse uso racional de energia.

Neste capítulo será apresentado o problema de perdas e modelizado no contexto das redes de energia eléctrica. Serão igualmente identificados os telecomandos que se utilizam para fazer alterações na rede.

2.2 Modelização do problema de perdas

O crescimento das redes de distribuição de energia eléctrica tem como consequência o aumento de perdas, uma vez que estas perdas variam consoante o número de barramentos e elementos (linhas, condensadores, transformadores) presentes na rede. Isto representa um elevado impacto para as empresas distribuidoras de energia eléctrica, pois parte da energia transportada é perdida.

As perdas de energia podem ser divididas em dois grupos [12, 20]:

- Perdas Técnicas:

Relacionadas com a transformação de energia eléctrica em energia térmica nos condutores (efeito *Joule*), perdas nos núcleos dos transformadores, perdas dieléctricas, perdas reactivas.

- Perdas Comerciais:

Associadas à energia que entrou na rede e que não foi facturada ao consumidor final. Estas perdas podem ocorrer devido ao furto de energia (ligações clandestinas à rede), adulterações no contador de energia, erros de medição e de facturação.

A complexidade deste problema é proporcional à extensão da rede, isto é, quanto maior o número de barramentos da rede maior será o problema, uma vez que as perdas de energia variam com o trânsito de energia na rede tal como explicado na secção 2.4. Nesta dissertação, é proposto reduzir as perdas técnicas através de alterações à topologia da rede existente.

Independentemente da rede de distribuição analisada, o problema da redução de perdas de energia pode ser definido segundo o seguinte modelo matemático [19]:

$$\begin{aligned} & \text{Minimizar} && p_{total} \\ & \text{s.a} && V_{min} \leq |V_i| \leq V_{max} \\ & && |I_{i,j}| \leq |I_{i,j \ max}| \end{aligned} \tag{2.1}$$

Onde

- p_{total} - valor de perdas totais do sistema
- $|V_i|$ - módulo da tensão no barramento i
- V_{min} e V_{max} - limites de tensão no barramento i
- $|I_{i,j}|$ - módulo da corrente eléctrica na linha entre o barramento i e j
- $|I_{i,j \ max}|$ - corrente eléctrica máxima admissível na linha

É de frisar que o valor total de perdas do sistema depende da topologia da rede, diagrama de carga e geração e das tensões dos barramentos, ou seja, do trânsito de energia.

2.3 Telecomandos em redes de distribuição

Uma vez estabelecida a rede, esta pode sofrer alterações em tempo-real através de acções realizadas por dispositivos de manobra. Estes dispositivos são estrategicamente colocados nas linhas de média e alta tensão, mais concretamente num ponto intermédio das linhas aéreas e instalados em apoios, proporcionando ao operador da rede um maior controlo sobre a mesma. A instalação de OCR's permite melhorar os tempos de interrupção e, consequentemente, aumentar a qualidade de serviço do sistema de energia eléctrica. Contudo, a instalação destes dispositivos tem um preço considerável, devendo proceder-se à selecção dos locais de instalação com critérios conducentes à optimização do seu custo-benefício. A instalação de OCR's permite melhorar o tempo de isolamento da falha, quando instalado em ramos Normalmente Fechados (NF), e o tempo de reconfiguração da falha, quando instalado em locais Normalmente Abertos (NA) [6].

Em Portugal, e mais concretamente na rede de distribuição explorada pela EDP, são utilizados os seguintes elementos [7, 14, 17]: IAT, IAR, OCR tipo 1 [9], OCR tipo 2 [10] e OCR tipo 3 [11].

A Figura 2.1 mostra um interruptor típico utilizado em linhas aéreas. Refira-se, que alguns dos aparelhos são controlados através de ondas de rádio e outros através da rede de comunicação existente na infraestrutura da rede eléctrica [21].

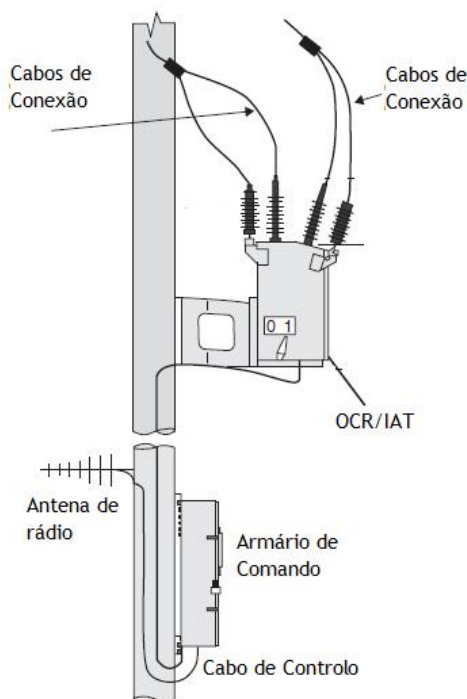


Figura 2.1: Exemplo de um Interruptor Aéreo Telecomandado [14]

O **Interruptor Aéreo Telecomandado** (IAT) é constituído por um órgão de corte e por um centro de comando. O centro de comando possibilita a supervisão e o controlo do IAT, tanto a nível local como remoto. A instalação deste aparelho tem como objectivo determinar se um troço ou uma área da rede tem uma avaria, recorrendo, apenas, a uma ordem de abertura, utilizando um mecanismo de inteligência que permite identificar o local do defeito. O seu princípio de funcionamento varia conforme a exploração da rede onde ele é instalado, podendo funcionar apenas como interruptor, permitindo efectuar manobras de reconfiguração da rede [7].

Outro tipo de aparelho é o **Interruptor Auto-Religador** (IAR), Figura 2.2. Este interruptor, tem como objectivo reduzir o tempo de localização de um defeito e o tempo de intervenção, melhorando assim a continuidade do serviço. Os IAR destinam-se à protecção de ramais em redes aéreas de média tensão. O princípio de funcionamento deste aparelho baseia-se na detecção da existência e ausência de tensão eléctrica durante o período de religações automáticas do disjuntor de protecção da linha onde se situa o respectivo IAR [7, 8].



Figura 2.2: Interruptor Auto-Religador em apoio de MT

O OCR1 (Figura 2.3), semelhante ao IAT, funciona como um interruptor-seccionador, podendo também ser fechado sobre correntes de curto-circuito, embora tenha uma capacidade inferior ao IAT/IAR, não devendo ser manobrado em condições de defeito. Porém, tem um tempo de manobra muito superior ao IAT/IAR. Este aparelho deve ser instalado nas fronteiras das linhas de MT, uma vez que não suporta altas correntes de curto-circuito [7, 9, 14, 17].



Figura 2.3: Órgão de Corte de Rede do Tipo 1 em apoio de MT

O OCR2 (Figura 2.4), é uma evolução do IAT, uma vez que pode actuar sobre correntes nominais e estabelecer correntes de curto-circuito. Este aparelho, difere do OCR1, pois tem a capacidade de suportar altas correntes de curto-circuito. O OCR2 possui uma distância entre pólos que não lhe confere isolamento e, na presença de sobretensões verifica-se escorvamento entre os pólos sendo necessário acrescentar um seccionador à montagem, permitindo que haja um corte visível. As suas características principais (poder

de corte e fecho, suporte de correntes de curto-circuito) fazem com que a sua instalação deva ser realizada de modo a isolar zonas ou linhas aéreas com defeito [7, 10, 14, 17].



Figura 2.4: Órgão de Corte de Rede do Tipo 2 em apoio de MT

A crescente preocupação com a continuidade do fornecimento de energia, nomeadamente com as interrupções de curta duração e a forma como estas podem afectar os clientes mais sensíveis, levou a EDP Distribuição a introduzir um novo equipamento para o telecomando de redes aéreas, denominado OCR3, sendo este o elemento mais recente a ser instalado na rede aérea MT nacional.

O **OCR3** (Figura 2.5) é um disjuntor-religador, que reúne as funções de todos os OCR's já existentes. Em relação ao OCR1 tem a possibilidade de executar manobras sobre defeito, e em relação ao OCR2 e OCR3 tem a capacidade de detectar a presença de tensão de ambos os lados dos terminais. Estas e outras características fazem com que uma rede com grande instalação deste tipo de aparelhos seja bastante automatizada. O OCR3 possui vários tipos de automatismo e deve ser coordenado com o disjuntor que se encontra na saída da subestação a montante. Pode operar em dois modos de funcionamento:

- No **modo recloser**, o OCR3 deve funcionar como disjuntor garantindo protecção de total da linha, permitindo a religação da mesma;
- No **modo seccionalizer**, o OCR3 deve funcionar como interruptor, tendo a capacidade de verificar a existência ou ausência de tensão e corrente na linha.

Normalmente os OCR3 são colocados no início de ramos com fraca qualidade de serviço e/ou na fronteira de redes urbanas ou industriais com redes rurais. Estes aparelhos permitem um escalonamento de troços de linhas extensas com cargas elevadas [7, 11, 14].



Figura 2.5: Órgão de Corte de Rede do Tipo 3 em apoio de MT

2.4 Avaliação de perdas

Trânsito de energia utilizando o Método de Gauss-Seidel

Trânsito de energia é a solução em regime estacionário de um sistema de energia eléctrica, compreendendo os geradores, a rede e as cargas.

O trânsito de energia compreende os seguintes passos [16]:

1. Formulação de um modelo matemático que represente com suficiente rigor o sistema físico real.
2. Especificação do tipo de barramentos e das grandezas referentes a cada um.
3. Solução numérica das equações do trânsito de energia, a qual fornece o valor das amplitudes e argumentos das tensões em todos os barramentos.
4. Cálculo das potências que transitam em todos os ramos, isto é, linhas e transformadores.

A Figura 2.6 vai servir de exemplo para o cálculo do trânsito de energia numa rede de 3 barramentos. A rede é constituída por dois barramentos de geração e um barramento de carga, duas linhas independentes e três cargas com diferentes elasticidades.

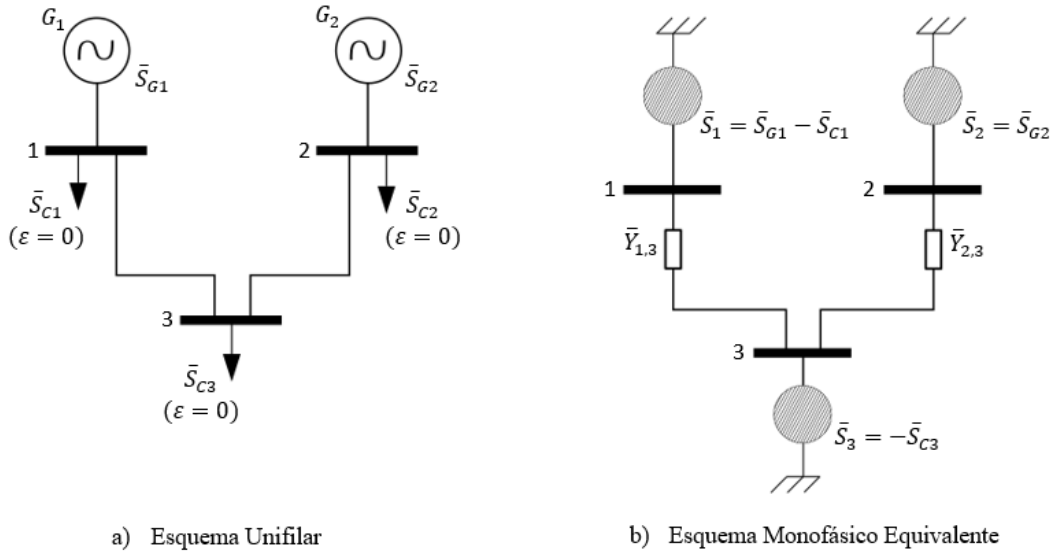


Figura 2.6: Sistema de 3 barramentos

Define-se potência injectada S como a diferença entre as potências gerada e consumida em cada barramento [16]:

$$S_i = S_{Gi} - S_{Ci} \quad (2.2)$$

Onde:

- S_i - Potência injectada no barramento i
- S_{Gi} - Potência gerada no barramento i
- S_{Ci} - Potência de carga no barramento i

Todavia, há que ter em consideração a elasticidade da carga (ε):

- Potência constante ($\varepsilon = 0$): a potência activa e reactiva não dependem da tensão;
- Corrente constante ($\varepsilon = 1$): a potência de carga é proporcional à amplitude da tensão;
- Admitância constante ($\varepsilon = 2$): a potência de carga é proporcional ao quadrado da amplitude da tensão.

Assim, da equação 2.2 vêm as seguintes potências nodais:

$$\begin{cases} S_1 = S_{G1} - S_{C1} \\ S_2 = S_{G2} \\ S_3 = -S_{C3} \end{cases}$$

Aplicando a primeira Lei de *Kirchoff* ou Lei das Correntes ao exemplo da figura 2.6, obtêm-se as seguintes equações nodais:

$$\begin{cases} \bar{I}_1 = \bar{I}_{1,3} \\ \bar{I}_2 = \bar{I}_{2,3} \\ \bar{I}_3 = \bar{I}_{1,3} + \bar{I}_{2,3} \end{cases} \Leftrightarrow \begin{cases} \frac{\bar{S}_1^*}{\bar{V}_1^*} = \bar{Y}_{1,3} \cdot \Delta \bar{V}_{1,3} \\ \frac{\bar{S}_2^*}{\bar{V}_2^*} = \bar{Y}_{2,3} \cdot \Delta \bar{V}_{2,3} \\ \frac{\bar{S}_3^*}{\bar{V}_3^*} = \bar{Y}_{1,3} \cdot \Delta \bar{V}_{1,3} + \bar{Y}_{2,3} \cdot \Delta \bar{V}_{2,3} \end{cases} \quad (2.3)$$

Onde:

- \bar{I}_i - Corrente injectada no barramento i , com $i=1,2,3$
- $\bar{I}_{i,j}$ - Corrente na linha entre os barramentos i e j
- \bar{S}_i^* - Potência complexa injectada no barramento i
- \bar{V}_i - Tensão complexa do barramento i
- $\bar{Y}_{i,j}$ - Admitância complexa na linha entre os barramentos i e j (no exemplo da figura 2.6 apenas foi considerado o modelo de linha longitudinal)
- $\Delta \bar{V}_{i,j}$ - Diferença entre as tensões dos barramentos i e j

O sistema de equações 2.3 pode ser escrito na forma matricial de acordo com a seguinte expressão:

$$[\bar{I}] = [\bar{Y}] \cdot [\bar{V}] \quad (2.4)$$

Onde:

- \bar{I} - Vector das correntes injectadas complexas
- \bar{Y} - Matriz das admitâncias nodais
- \bar{V} - Vector das tensões nodais complexas

Assim, aplicando a expressão 2.4 ao sistema 2.3 ficamos com a seguinte forma matricial:

$$\begin{bmatrix} \frac{\bar{S}_1^*}{\bar{V}_1^*} \\ \frac{\bar{S}_2^*}{\bar{V}_2^*} \\ \frac{\bar{S}_3^*}{\bar{V}_3^*} \end{bmatrix} = \begin{bmatrix} \bar{Y}_{1,1} & \bar{Y}_{1,2} & \bar{Y}_{1,3} \\ \bar{Y}_{2,1} & \bar{Y}_{2,2} & \bar{Y}_{2,3} \\ \bar{Y}_{3,1} & \bar{Y}_{3,2} & \bar{Y}_{3,3} \end{bmatrix} \cdot \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \\ \bar{V}_3 \end{bmatrix} \quad (2.5)$$

O sistema anterior, na topologia da figura 2.6 fica:

$$\begin{bmatrix} \frac{\bar{S}_1^*}{\bar{V}_1^*} \\ \frac{\bar{S}_2^*}{\bar{V}_2^*} \\ \frac{\bar{S}_3^*}{\bar{V}_3^*} \end{bmatrix} = \begin{bmatrix} \bar{Y}_{1,1} & 0 & \bar{Y}_{1,3} \\ 0 & \bar{Y}_{2,2} & \bar{Y}_{2,3} \\ \bar{Y}_{3,1} & \bar{Y}_{3,2} & \bar{Y}_{3,3} \end{bmatrix} \cdot \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \\ \bar{V}_3 \end{bmatrix} \quad (2.6)$$

É de notar que estas equações relacionam tensões e potências, e não correntes, o que as torna não-lineares. Daí ser necessário recorrer a métodos como o de *Gauss-Seidel*.

Da forma matricial apresentada na equação 2.6 retiramos que a formula para calcular as tensões nodais é:

$$\begin{cases} \frac{\bar{S}_1^*}{\bar{V}_1^*} = \bar{Y}_{1,1} \cdot \bar{V}_1 + \bar{Y}_{1,3} \cdot \bar{V}_3 \\ \frac{\bar{S}_2^*}{\bar{V}_2^*} = \bar{Y}_{2,2} \cdot \bar{V}_2 + \bar{Y}_{2,3} \cdot \bar{V}_3 \\ \frac{\bar{S}_3^*}{\bar{V}_3^*} = \bar{Y}_{3,1} \cdot \bar{V}_1 + \bar{Y}_{3,2} \cdot \bar{V}_2 + \bar{Y}_{3,3} \cdot \bar{V}_3 \end{cases} \quad (2.7)$$

Da equação 2.7 retiramos as equações para o cálculo das tensões nodais:

$$\begin{cases} V_1^k = \frac{1}{y_{1,1}} \cdot \left(\frac{S_1^*}{(V_1^{k-1})^*} - y_{1,3} \cdot V_3^{k-1} \right) \\ V_2^k = \frac{1}{y_{2,2}} \cdot \left(\frac{S_2^*}{(V_2^{k-1})^*} - y_{2,3} \cdot V_3^{k-1} \right) \\ V_3^k = \frac{1}{y_{3,3}} \cdot \left(\frac{S_3^*}{(V_3^{k-1})^*} - y_{3,1} \cdot V_1^k - y_{3,2} \cdot V_2^k \right) \end{cases} \quad (2.8)$$

A sequência do processo de cálculo dos valores das tensões nodais inicia-se com uma estimativa do valor inicial das amplitudes e argumentos dessas mesmas tensões. Completada a primeira iteração, o processo repete-se por iterações sucessivas, até que o valor complexo das tensões nodais cumpra o requisito de precisão definido, sendo este, normalmente, definido por $\varepsilon = 10^{-4}$. Perante estas condições, o método é convergente, e a solução do mesmo é mais rigorosa quanto maior for o número de iterações.

Uma vez calculadas as tensões nodais através do trânsito de energia podemos então substituir os seus valores na equação 2.7 e calcular o valor das potências nodais.

As perdas são calculadas em função da geração, da topologia e da carga. Tendo obtido o valor das tensões nodais, é possível, finalmente, calcular as perdas transitadas numa linha através das equações:

$$S_{ij} = P_{ij} + jQ_{ij} = V_i \frac{V_i^* - V_j^*}{R_{ij} - jX_{ij}} - j \frac{y_{ij}^*}{2} V_i^2 \quad (2.9)$$

$$S_{ji} = P_{ji} + jQ_{ji} = V_j \frac{V_j^* - V_i^*}{R_{ji} - jX_{ji}} + \frac{y_{ij}^*}{2} V_j^2 \quad (2.10)$$

Assim, somando as equações 2.9 e 2.10 é possível calcular a potência transitada na linha:

$$\bar{S}_{ij} + \bar{S}_{ji} = P_{perdas_{ij}} + jQ_{perdas_{ij}} \quad (2.11)$$

Somando os valores de todas as potências transitadas nas linhas obtemos através das equações 2.12 e 2.13 o total de perdas numa rede.

$$P_{p_{total}} = \sum P_{perdas} \quad (2.12)$$

$$Q_{p_{total}} = \sum Q_{perdas} \quad (2.13)$$

Embora o método de *Gauss-Seidel* tenha sido o primeiro algoritmo utilizado para resolver problemas de trânsito de energia, este foi ultrapassado pelo método de *Newton-Raphson* devido à superior velocidade de convergência. Assim, para efeitos de simulação é utilizado o método de *Newton-Raphson* uma vez que é mais rápido a convergir.

MÉTODOS DE RECONFIGURAÇÃO TOPOLÓGICA DE REDES PARA REDUÇÃO DE PERDAS

3.1 Definição da metodologia

Diversos autores abordam o tema da redução de perdas e a colocação de OCR's em várias redes de distribuição, propondo diferentes metodologias que abordam o problema em questão. Os métodos propostos, incluem algoritmos genéticos [4, 22], métodos iterativos [19] e decomposição e análise convexa [6].

De forma a definir a base do problema e qual a metodologia a aplicar para a redução de perdas, foi feito um levantamento de vários artigos bibliográficos dos quais foram seleccionados três artigos científicos e uma dissertação.

1. No **artigo** [19] é apresentado um algoritmo de reconfiguração para a optimização de um sistema de distribuição radial. Essa reconfiguração exige que seja feita uma selecção das melhores linhas a serem abertas de forma a que o sistema de distribuição obtenha a *performance* desejada. O algoritmo proposto é baseado num conjunto simples de regras heurísticas que identificam o conjunto de configurações que resultem no mínimo de perdas do sistema de distribuição. Este algoritmo é dividido em duas partes: uma **primeira** onde é determinada a melhor combinação de alterações com o menor esforço computacional possível; na **segunda** parte, é feita uma optimização do cálculo de perdas da configuração encontrada na primeira parte através do trânsito de energia.
2. Na **dissertação** [17], o autor baseando-se no artigo anterior, propôs através da colocação de órgãos de corte da rede em pontos estratégicos diminuir as perdas e melhorar a fiabilidade de uma rede de distribuição de energia eléctrica. Para tal,

desenvolveu um algoritmo semelhante ao do artigo [19] e aplicou-o a um caso real - Rede de Santiago, Cabo Verde.

3. Por sua vez, no **artigo** [22] é proposto um método de optimização que identifica o posicionamento ideal de OCR's numa rede de distribuição com vários pontos de geração. A função objectivo é definida através de um índice de fiabilidade do serviço, sabendo que ao haver várias subestações é possível reduzir bastante o número de interrupções e a duração das mesmas. Assim, é utilizado um algoritmo genético de população múltipla para encontrar as melhores soluções. Através da utilização deste algoritmo, a optimização pode ser resolvida utilizando a população geral, e a influência de parâmetros genéticos inadequados pode ser reduzida e a convergência prematura pode ser também superada com eficácia.
4. No **artigo** [4] é, também, proposto um modelo matemático para a localização ideal de religadores automáticos em sistemas de distribuição para isolamento de falhas e transferência de carga. Como primeiro objectivo, são considerados os níveis de energia do sistema juntamente com a fiabilidade dos sistemas de comunicação dos religadores, e como segundo objectivo, são considerados os custos de investimento desses mesmo religadores e da tecnologia de comunicação que eles possuem. Para resolução do problema, é utilizado um micro-algoritmo genético porque se revelam bastante eficientes em termos computacionais.

Nesta dissertação foi inicialmente desenvolvido, em *Python*, um método de busca sequencial baseado na dissertação [17] com o objectivo de obter melhores resultados e com um tempo de computação mais rápido. Posteriormente, foi desenvolvido um Algoritmo Genético, para resolver o mesmo problema, uma vez que se verificou em vários artigos que este poderia trazer diversificação às soluções e melhoria das mesmas. O AG é baseado no conceito clássico do mesmo porém, devido às características do problema foi necessário realizar algumas alterações aos vários operadores genéticos, como será explicado no capítulo 3.3.

3.2 Método de busca sequencial

A metodologia proposta vai muito ao encontro da metodologia proposta na dissertação [17] apresentando algumas particularidades devido à utilização directa do *software* PSS/E para o cálculo do trânsito de energia.

Na Figura 3.1 encontra-se representado o fluxograma que contém os passos do método sequencial proposto para a resolução do problema de perdas.

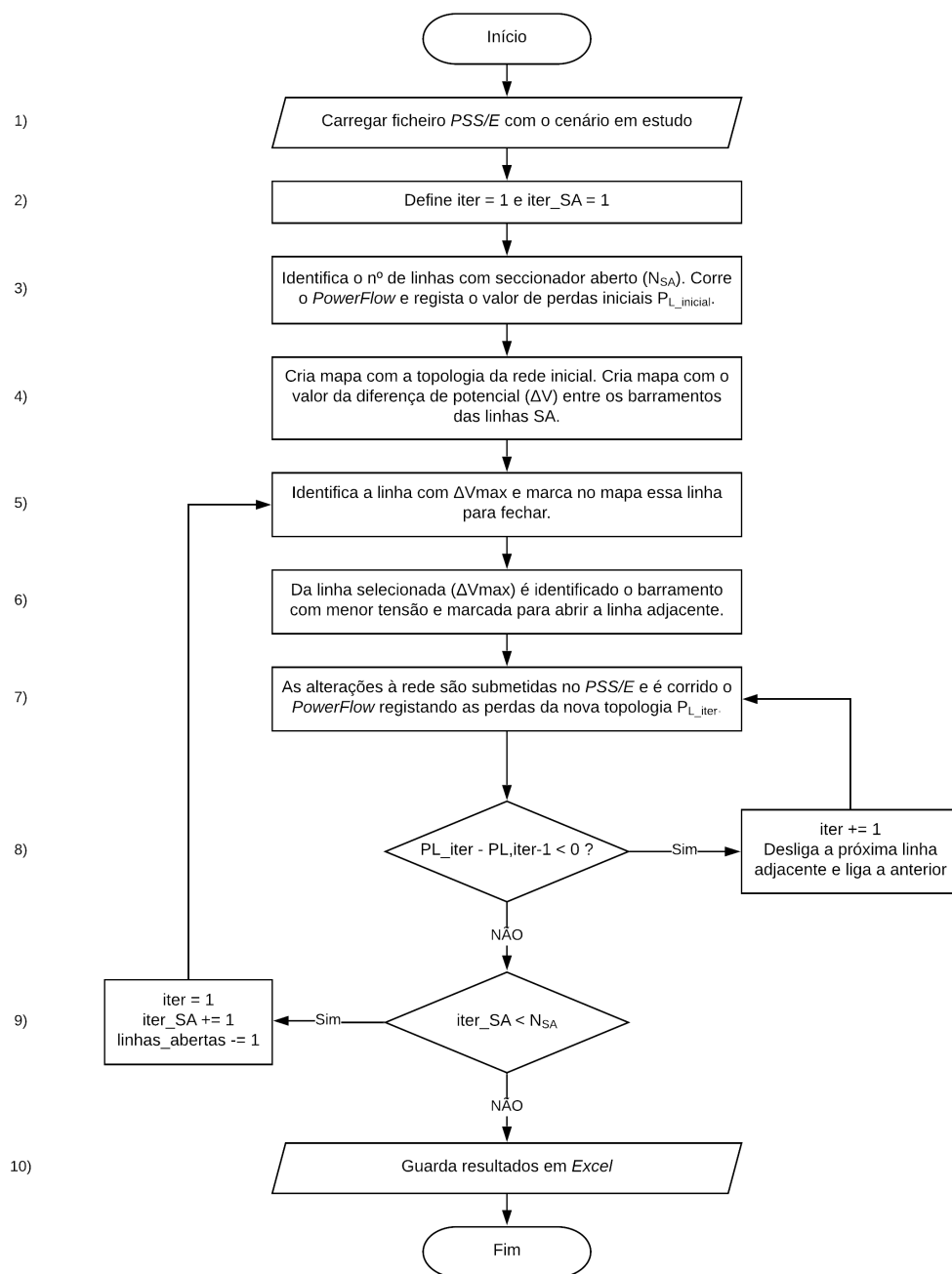


Figura 3.1: Fluxograma do método de busca sequencial em *Python*

O método proposto apresenta os seguinte passos:

- 1) Carrega o ficheiro *PSS/E* que contém o cenário em estudo. Neste cenário estão todas as características técnicas das linhas e de outros equipamentos (baterias de condensadores, transformadores) e os diagramas de carga e geração;
- 2) Inicia os contadores de iterações ($iter = 1$) e o contador do número de operações feitas a linhas com seccionador aberto ($iter_SA = 1$);

- 3) Identifica o número de linhas SA e guarda esse mesmo valor (N_{SA}). Corre o trânsito de energia e regista o valor de perdas inicial da rede em análise;
- 4) Cria um mapa (matriz) com $N_{barramentos} \cdot N_{barramentos}$ que contém a topologia da rede. Cria uma segunda matriz com a mesma dimensão que contém o valor da diferença de potencial entre os barramentos que contém uma linha com seccionador aberto. O código em *Python* referente ao desenvolvimento deste passo encontra-se no ponto 1 no anexo I.1.
- 5) Identifica qual das linhas com SA tem ΔV_{max} e assinala no mapa o fecho dessa linha;
- 6) Das extremidades da linha com SA é identificado o barramento com menor valor de tensão e é marcada para abrir a linha adjacente;
- 7) As alterações à rede são submetidas para o PSS/E e é corrido o trânsito de energia registando as perdas da nova topologia P_{iter} ;
- 8) Verifica a condição $P_{iter} - P_{iter-1} < 0$:
 - a) Se verdadeira, incrementa o contador de iterações e abre a próxima linha adjacente e fecha a anterior;
 - b) Se falsa, executa o passo 9);
- 9) Verifica a condição $iter_{SA} < N_{SA}$:
 - a) Se verdadeira, dá *reset* ao contador de iterações, incrementa o contador de operações SA e decrementa o contador N_{SA} e a execução regressa ao passo 5);
 - b) Se falsa, executa o passo 10);
- 10) Apresenta os resultados obtidos ao longo da execução do programa e a configuração da rede otimizada para o problema de redução de perdas. Interrompe a execução do programa.

No ponto 2 do anexo I.1, é possível encontrar as funções essenciais do código em *Python* que permitem executar os passos 5) a 7).

De forma a demonstrar o funcionamento do método apresentado, foi desenvolvida uma pequena rede que se encontra na Figura 3.2. Esta rede é constituída por 1 subestação, 8 barramentos e 9 linhas das quais 2 são linhas com seccionador aberto. No anexo II.1 é possível encontrar toda a informação relativa às características da rede utilizada para a demonstração.

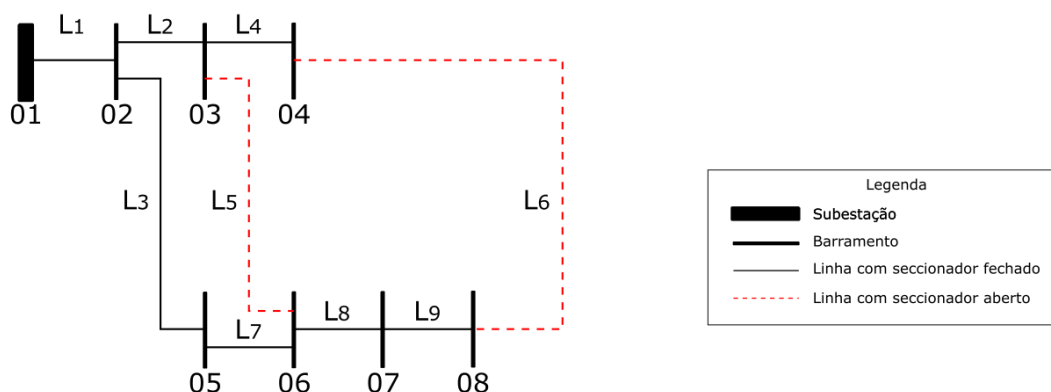


Figura 3.2: Rede ilustrativa - Método de busca sequencial

Em seguida, são apresentados passo a passo os itens mais relevantes do processo:

- É calculado o trânsito de energia, os valores podem ser analisados no anexo II.2 coluna 'Iteração: Inicial';
- É calculada a queda de tensão entre as linhas com seccionador aberto;

$$\text{Linha n}^\circ 5 \rightarrow \Delta V_{03,06} = |\bar{V}_{03} - \bar{V}_{06}| = |0,9891 - 0,9789| = 0,0102pu$$

$$\text{Linha n}^\circ 6 \rightarrow \Delta V_{04,08} = |\bar{V}_{04} - \bar{V}_{08}| = |0,9884 - 0,9743| = 0,0141pu$$

- A linha n° 6 é seleccionada para ligar, uma vez que, apresenta a maior diferença de tensão entre dois barramentos, e para desligar é seleccionada a linha n° 9, pois é a linha adjacente ao barramento de menor tensão (\bar{V}_{08}) e também mantém a rede radial. A nova configuração é a seguinte:

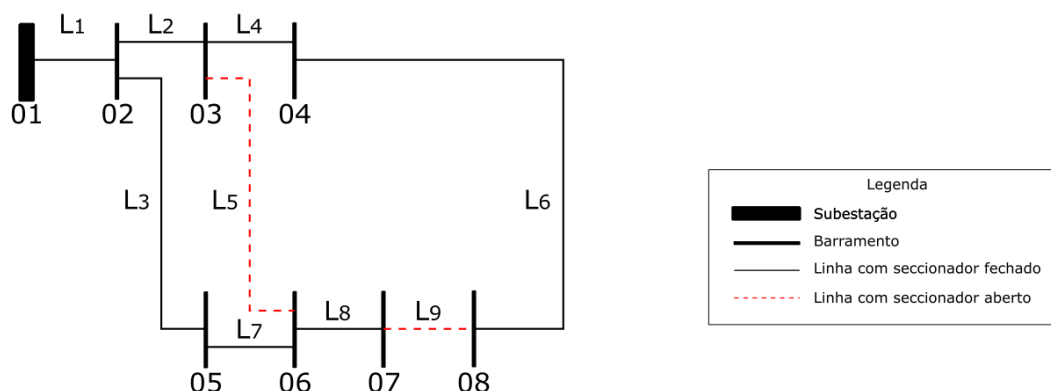


Figura 3.3: Rede ilustrativa - Método de busca sequencial (Iteração n° 1)

- É novamente calculado o trânsito de energia, anexo II.2 coluna 'Iteração: 1';

e) Verifica-se se houve redução de perdas;

$$\Delta p_{activas}^{inicial-1} = p_{activas}^{inicial} - p_{activas}^1 = 0,1107 - 0,1005 = 0,0102pu$$

f) Seleccionar para ligar linha nº09 e para desligar a linha nº 8, porque é a próxima linha adjacente à linha anteriormente seleccionada. A nova configuração é a seguinte:

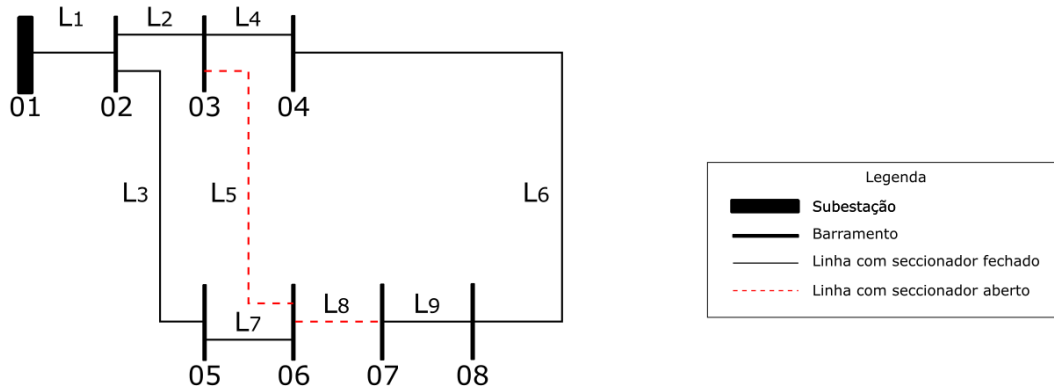


Figura 3.4: Rede ilustrativa - Método de busca sequencial (Iteração nº 2)

g) É novamente calculado o trânsito de energia, anexo II.2 coluna 'Iteração: 3';

h) Verifica-se se houve redução de perdas;

$$\Delta p_{activas}^{1-2} = p_{activas}^1 - p_{activas}^2 = 0,1005 - 0,1081 = -0,0076pu$$

Nesta iteração, verificou-se de que não houve melhoria de perdas. Logo a configuração da Figura 3.3 será a base para a próxima iteração.

Assim, voltando ao passo b), com base nos resultados da segunda iteração é calculada a queda de tensão na linha nº 5. Foram verificadas redução de perdas nas iterações seguintes à semelhança do indicado nos passos anteriores.

i) Concluídas todas as iterações possíveis, foi obtida a configuração final:

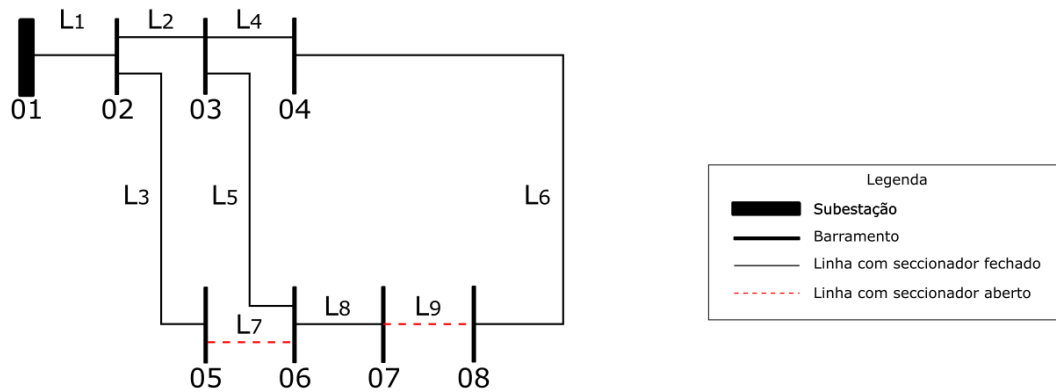


Figura 3.5: Rede ilustrativa - Método de busca sequencial (Configuração final)

Alterando a configuração inicial da rede, foi possível verificar que ligando as linhas nº 5 e nº 6, e desligando as linhas nº 7 e nº 9 as perdas diminuíram de 110,7 kW para 108,9 kW.

3.3 Algoritmo Genético

Como foi modelizado no capítulo 2.2, o problema de perdas encontra-se dependente da dimensão da rede. Sendo o problema combinatório e discreto, os AG têm de ser adaptados a este tipo de problemas de optimização de forma a abranger o maior número de possibilidades existentes na rede.

Um AG não é mais do que uma procura heurística inspirada na teoria da evolução natural apresentada por Charles Darwin.

Na Figura 3.6 vemos representado os vários elementos que são a base para o conceito de um algoritmo genético. A população é constituída por vários cromossomas que por sua vez são formados por um conjunto de genes.

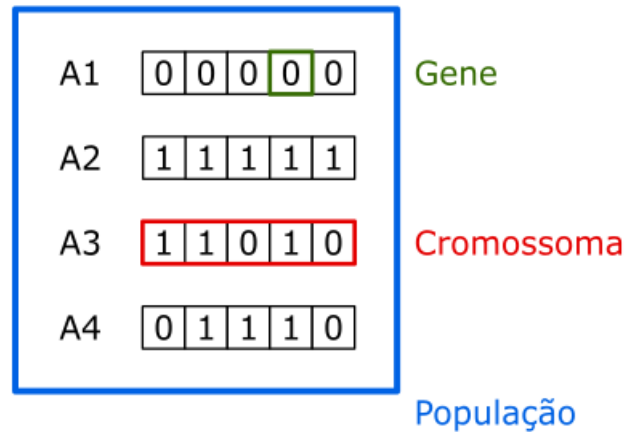


Figura 3.6: Esquema ilustrativo dos elementos de um algoritmo genético

O processo de selecção natural começa com a escolha do indivíduo com a melhor função objectivo, ou *fitness*, de uma população. Após a selecção, os indivíduos escolhidos vão produzir filhos que herdam as suas características e estes serão adicionados à próxima geração. Quanto maior for a *fitness* dos pais, melhor será a dos seus filhos nas futuras gerações tendo estes maior chance de sobreviver. Este processo continua até final sendo o resultado uma população que contém indivíduos com a melhor *fitness* possível.

Tipicamente, um algoritmo genético considera as seguintes fases:

- Inicialização:

Geração de população inicial usando codificação adequada.

- Função-objectivo ou *Fitness*:

É o objectivo da optimização do problema. Nesta dissertação, o objectivo é a optimização das perdas de energia numa rede eléctrica. A grande vantagem dos AG's está no facto de não ser preciso saber como funciona a função objectivo, apenas é preciso tê-la disponível para ser aplicada aos indivíduos e comparar resultados.

- Seleção:

Representa a seleção dos indivíduos aleatoriamente ou de acordo com a sua função-objectivo.

- Cruzamento ou *Crossover*:

Operador utilizado para manter a diversidade da população.

- Mutação:

Operador utilizado para manter a diversidade da população.

Uma vez que o problema apresentado apresenta particularidades que não poderiam ser resolvidos utilizando apenas o método tradicional, diversas alterações foram aplicadas sobre os operadores genéticos de forma a ser possível respeitar as condições do problema.

O algoritmo garante que as seguintes condições são respeitadas:

1. Radialidade:

A rede tem de ser radial, não podendo haver *loops*, isto é, não pode ser possível passando por um barramento regressar a esse mesmo barramento;

2. Não podem existir 'ilhas':

Todas as soluções têm de garantir de que não existem barramentos que ficam sem alimentação;

3. Sobreensão:

Nenhuma das linhas da rede pode estar em sobreensão.

O fluxograma apresentado na Figura 3.7 permite esquematizar o funcionamento do algoritmo genético.

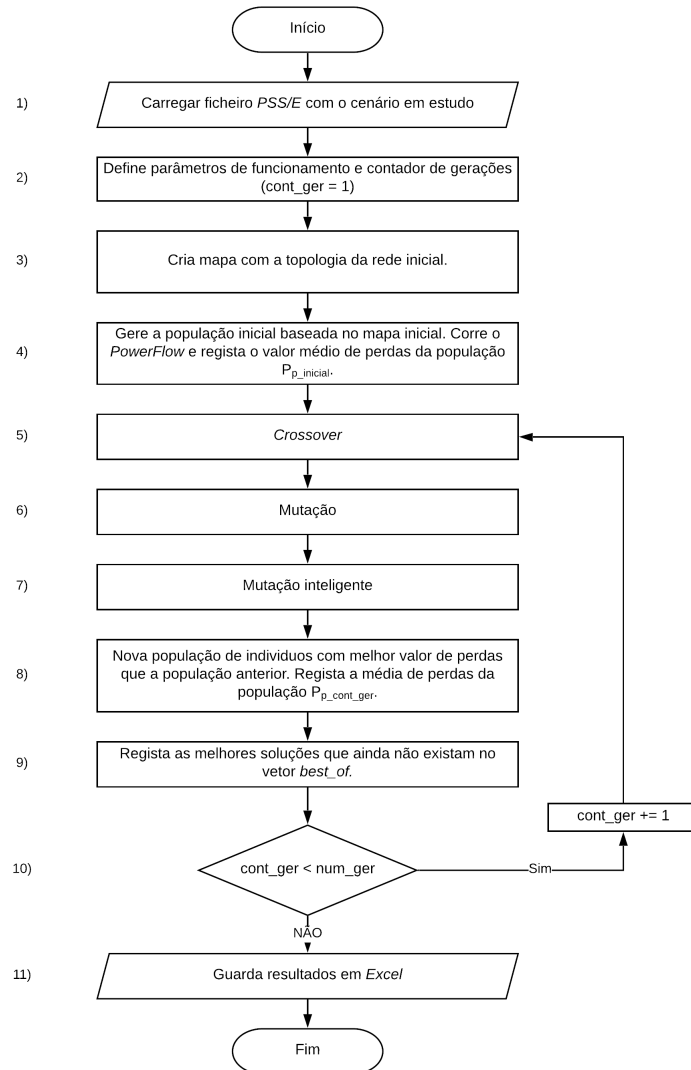


Figura 3.7: Fluxograma do algoritmo gen tico desenvolvido em *Python*

O algoritmo gen tico proposto apresenta os seguinte passos:

- 1) Carrega o ficheiro PSS/E que cont m o cen rio em estudo. Neste cen rio est o todas as caracter sticas t cnicas das linhas e de outros equipamentos (baterias de condensadores, transformadores) e os diagramas de carga e gera  o;
- 2) Define par metros de funcionamento (tamanho da popula  o, n mero de gera  es e probabilidade de muta  o simples e inteligente). Inicia o contador de gera  es ($cont_ger = 1$);
- 3) Cria um mapa (matriz) com $N_{barramentos} \times N_{barramentos}$ que cont m a topologia da rede inicial;
- 4) Gere a popula  o inicial de forma aleat ria a partir da topologia inicial.   corrido o tr nsito de energia e registado o valor m dio de perdas da popula  o inicial ($P_{p_inicial}$)

- 5) *Crossover*;
- 6) Mutação simples;
- 7) Mutação inteligente;
- 8) Nova população de indivíduos que evoluiu a partir da população que lhe deu origem. Regista a média de perdas da nova população ($P_{p_num_ger}$);
- 9) Regista as melhores soluções que ainda não existem num vetor *best off*;
- 10) Verifica a condição $cont_ger < num_ger$:
 - a) Se verdadeira, incrementa o contador de gerações e continua a execução a partir do passo 5);
 - b) Se falsa, executa o passo 11);
- 11) Apresenta os resultados obtidos ao longo da execução do programa e a configuração da rede optimizada para o problema de redução de perdas. Interrompe a execução do programa.

3.3.1 Codificação e Inicialização da população

Uma vez que o problema apresentado difere dos problemas clássicos resolvidos pelos AG's, foi necessário adaptar uma solução que conseguisse codificar a topologia de uma rede de distribuição. Assim, foi criada uma matriz com dimensão variável, $N_{barramentos} \cdot N_{barramentos}$, sendo daqui em diante denominada por **mapa**.

Utilizando novamente como exemplo ilustrativo a rede da Figura 3.8 encontra-se na Figura 3.9 o mapa que contém a codificação dessa mesma rede.

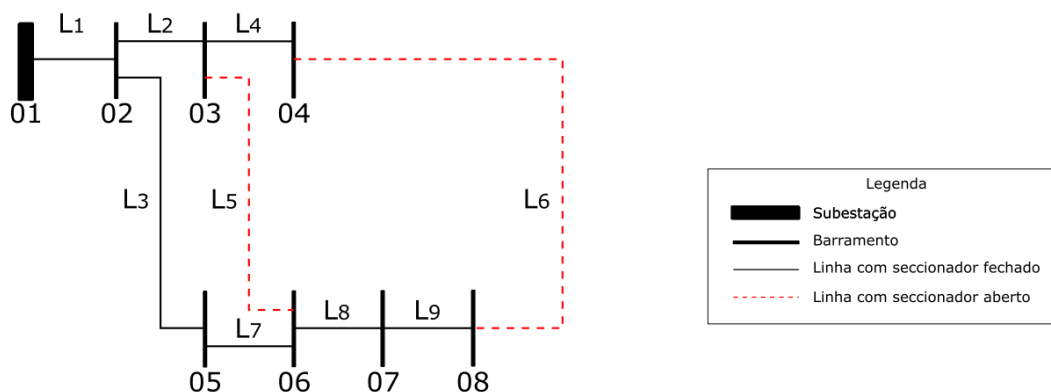


Figura 3.8: Rede ilustrativa - Algoritmo Genético

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	2	0	0	0	0	0	0
2	0	2	0	2	0	2	0	0	0
3	0	0	2	0	2	0	1	0	0
4	0	0	0	2	0	0	0	0	1
5	0	0	2	0	0	0	2	0	0
6	0	0	0	1	0	2	0	2	0
7	0	0	0	0	0	0	2	0	2
8	0	0	0	0	1	0	0	2	0

Legenda

0 - não existem elementos (linhas ou transformadores)

1 - existe linha com seccionador aberto (SA)

2 - existe linha com seccionador fechado (SF)

Figura 3.9: Codificação da rede da Figura 3.8

Fazendo a ligação entre o problema e o esquema apresentado na Figura 3.6, uma população é um conjunto de mapas, cada mapa é um cromossoma e cada valor do mapa é um gene.

A nível de código, todos os valores do mapa são inicialmente preenchidos por zeros. Posteriormente, o valor de cada posição é comparado com a informação contida no ficheiro *PSS/E* e alterado de acordo com a regra apresentada na Figura 3.9.

Para além da codificação da rede, foi necessário codificar um vector com N_{SA} posições, que contém a configuração a aplicar às linhas com SA. A Figura 3.10 serve como exemplo, sendo possível retirar a informação de que a 1ª, 2ª e 4ª linha estão marcadas para fechar, mantendo-se as restantes abertas.

1	1	0	1	0	Vector SA
---	---	---	---	---	-----------

Figura 3.10: Exemplo de configuração para um vector que representa as linhas com SA

Esta última codificação é utilizada para a criação de novos membros sendo que estes apenas são criados para preencher a população inicial.

Para criar novos membros gera-se N novos mapas até atingir o número de membros pretendidos. Para gerar esses mapas é gerada aleatoriamente a configuração do vector SA. Esta configuração vai ser aplicada ao mapa original que foi carregada do ficheiro *PSS/E*. Uma vez fechada uma das linhas com SA, é necessário abrir aleatoriamente uma das linhas com SF, sendo este processo repetido até o vector SA ser todo ele percorrido. Cada configuração é testada e se cumprir as condições de radialidade é adicionada à população inicial.

3.3.2 Cruzamento

Um dos maiores desafios encontrados no desenvolvimento do algoritmo deu-se na fase do cruzamento, ou *crossover*, de membros.

Na Figura 3.11 está representado o cruzamento clássico entre os dois primeiros membros do exemplo apresentado na Figura 3.6.

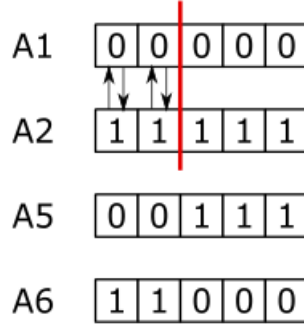


Figura 3.11: Representação do funcionamento do *crossover*

Para criar novos indivíduos a partir dos membros já existentes é necessário haver cruzamento entre eles. Todavia, enquanto o *crossover* clássico, representado na figura anterior, escolhe aleatoriamente uma posição de um cromossoma, parte-o em dois e cruza com outro cromossoma, no problema apresentado isso não é possível devido às condições que têm de ser respeitadas.

Assim, para resolver este obstáculo, foi proposto o cruzamento entre os indivíduos de forma comparativa, isto é, compara-se entre os dois membros se onde um tem uma linha aberta o outro tem a mesma linha fechada, verificando a seguinte condição:

$$membro1[x, y] = 1 \ \& \ membro2[x, y] = 2 \quad (3.1)$$

Uma vez verificada a condição 3.1, o novo individuo vê a linha $[x, y]$ alterada de aberta para fechada. Posteriormente, e de forma a manter a radialidade é testada a condição inversa:

$$membro1[x, y] = 2 \ \& \ membro2[x, y] = 1 \quad (3.2)$$

O novo individuo vê novamente a sua configuração alterada, neste caso de fechada para aberta. Depois de efectuadas as alterações a solução é testada. Se for válida, o cruzamento dá-se por concluído com sucesso e o novo membro é adicionado à população. Caso contrário, a configuração é resposta e é procurada a próxima linha que verifique a condição 3.2 até ser obtida uma solução válida. Se a segunda condição nunca se verificar é encontrada a linha seguinte que verifica a condição 3.1 e o processo é repetido.

A6	1	1	0	0	0
A6*	1	1	0	0	1

Figura 3.15: Exemplo de mutação de um indivíduo

A mutação ocorre durante o processo de evolução da nova população, como está representado na figura anterior, e de acordo com uma probabilidade definida pelo utilizador. Esta deve ser baixa, pois se for muito elevada o algoritmo tornar-se-à numa procura aleatória primitiva.

Vários tipos de mutação podem ser encontrados no artigo [18], porém, e novamente, perante a problemática da radialidade foi necessário desenvolver um processo de mutação diferente do conceito clássico.

Assim, o perante a probabilidade de ocorrer mutação é escolhida aleatoriamente uma linha com SA para fechar. Uma vez fechada, é escolhida aleatoriamente uma linha com SF para abrir e testada a solução. Se não for válida escolhe-se novamente uma linha para abrir até a solução ser válida. Se em cinco tentativas não for encontrada solução, é escolhida aleatoriamente outra linha com SA para fechar e repete-se o processo. A esta mutação foi chamada **mutação simples**.

A Figura 3.16, permite ilustrar o que acontece numa mutação simples. A configuração da rede foi alterada aleatoriamente, fechando a linha 9 e abrindo a linha 6.

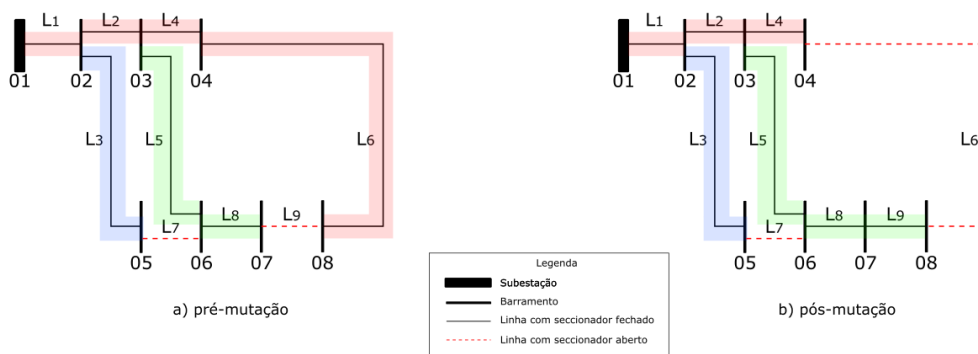


Figura 3.16: Mutação simples - Rede exemplo

Com visto a atingir soluções ainda melhores, foi desenvolvido outro tipo de mutação para este problema, denominada de **mutação inteligente**.

A mutação inteligente tem como base a mutação simples com a particularidade de que nesta é escolhida a melhor das mutações, isto é, uma vez escolhida a linha com SA para fechar, todas as linhas com SF são abertas e testadas uma a uma e as soluções que forem válidas são guardadas. No final, a *fitness* dos resultados válidos são comparadas e é

escolhido o indivíduo com melhor valor de *fitness* para passar à próxima fase do processo evolutivo.

O fluxograma da Figura 3.17 ilustra o procedimento do operador genético. O código em *Python* respectivo ao desenvolvimento deste procedimento encontra-se no ponto 2 do anexo I.2.

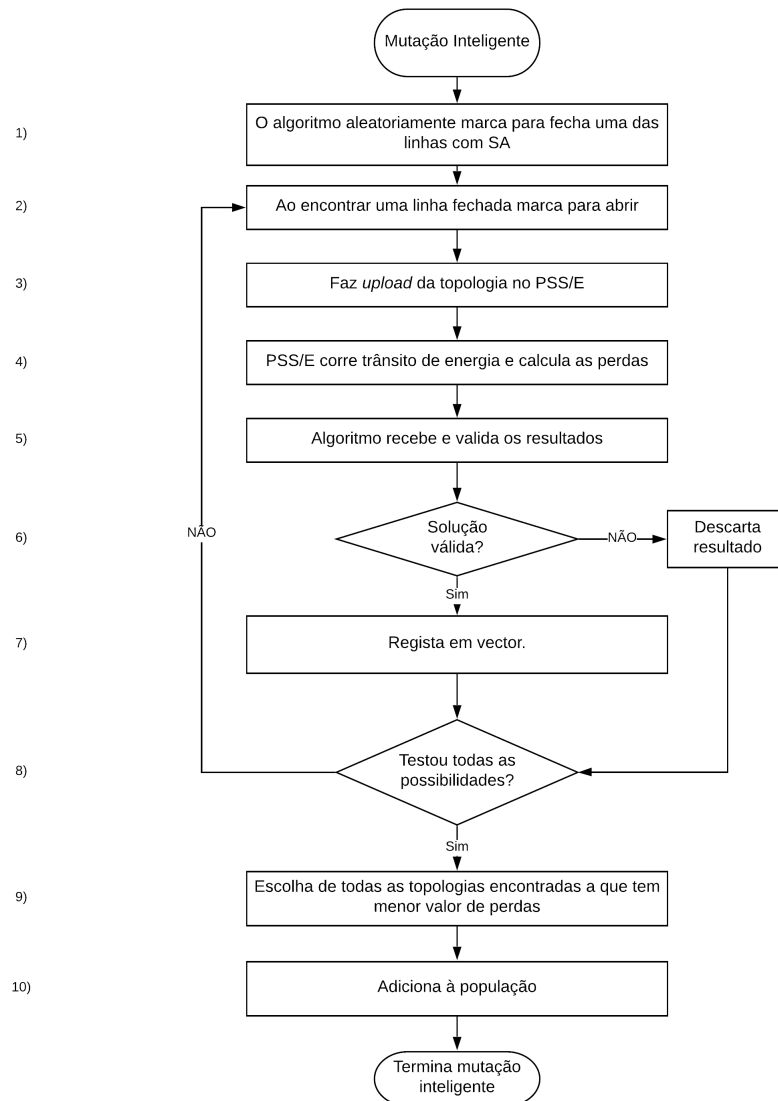


Figura 3.17: Fluxograma Mutação Inteligente

Na Figura 3.18, observamos a configuração inicial que irá sofrer uma mutação inteligente. A *fitness* desta configuração é de 145,2 kW.

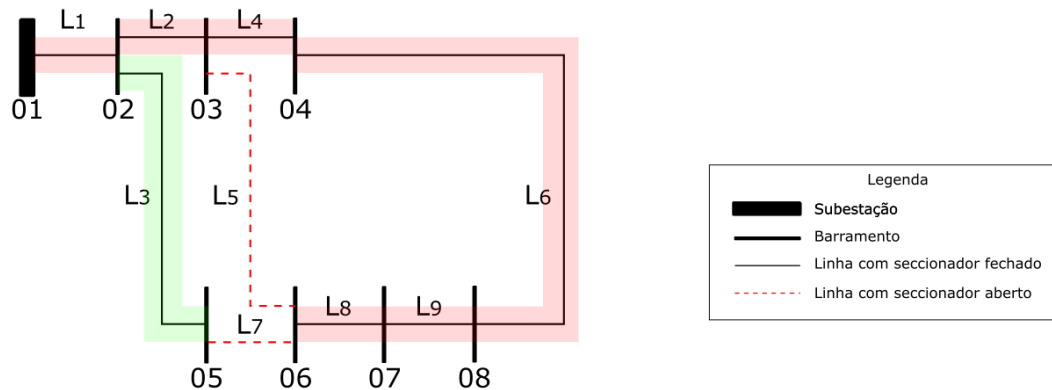


Figura 3.18: Mutação inteligente - Configuração inicial

A rede da figura anterior representa o indivíduo que foi aleatoriamente seleccionado para entrar na mutação inteligente. Da mesma forma, aleatória, foi escolhida para fechar um das linhas com Seccionador Aberto, isto é, a linha 5 ou a 7, sendo neste caso escolhida para fechar a linha 5. Em seguida, todas as linhas que se encontram fechadas (excepto a linha 5) irão ser abertas e testadas, estando demonstradas nas próximas figuras todas as configurações válidas.

Na primeira configuração, Figura 3.19, foi aberta a linha 4 e o valor da *fitness* é de 125,5 kW.

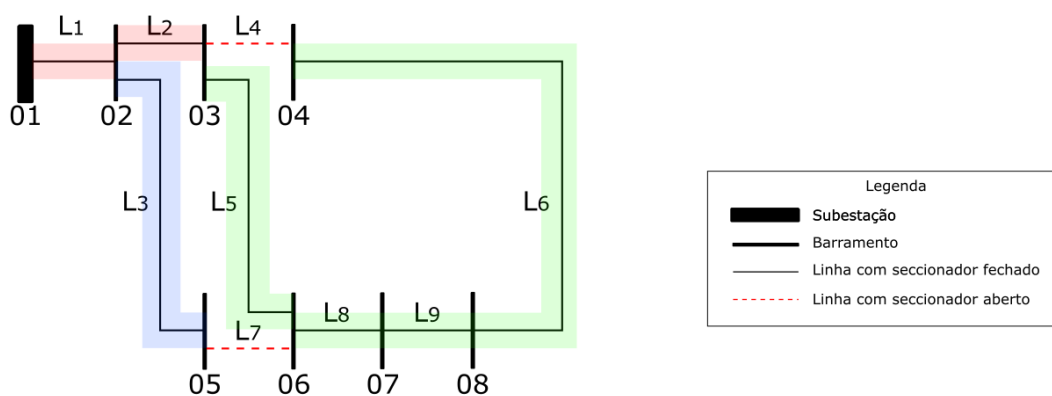


Figura 3.19: Mutação inteligente - 1ª configuração

Na primeira configuração, Figura 3.20, foi aberta a linha 6 e o valor da *fitness* é de 115,6 kW.

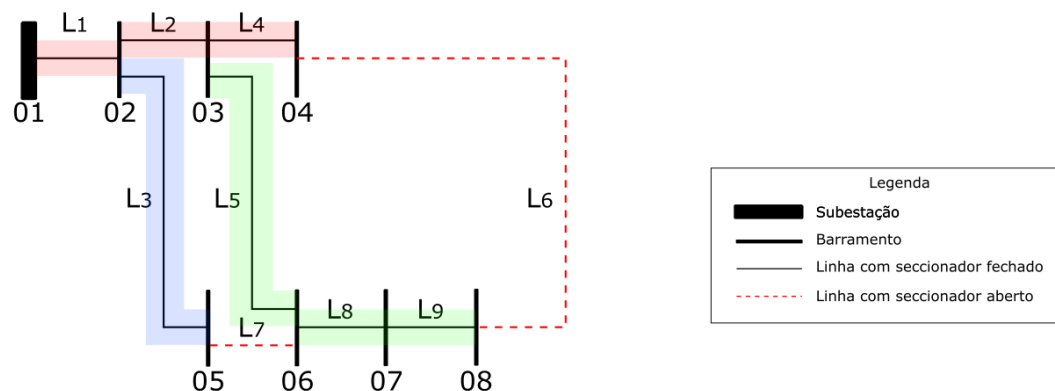


Figura 3.20: Mutação inteligente - 2ª configuração

Na primeira configuração, Figura 3.21, foi aberta a linha 8 e o valor da *fitness* é de 117,3 kW.

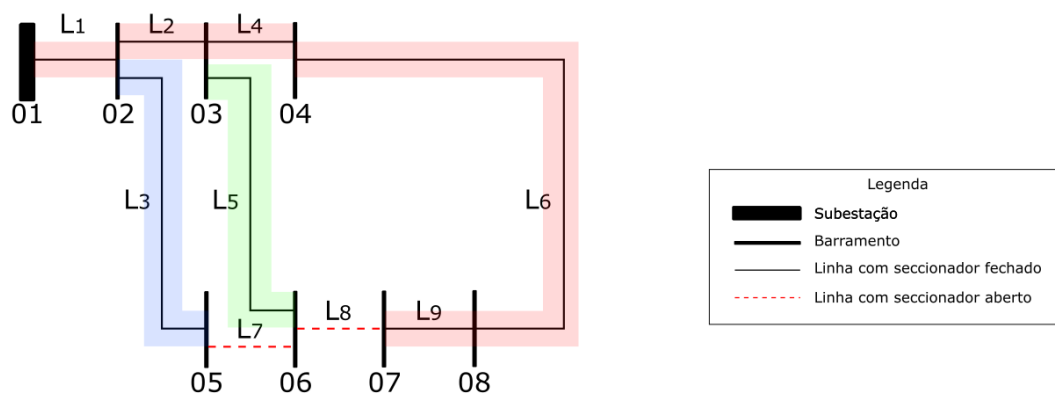


Figura 3.21: Mutação inteligente - 3ª configuração

Na primeira configuração, Figura 3.22, foi aberta a linha 9 e o valor da *fitness* é de 108,9 kW.

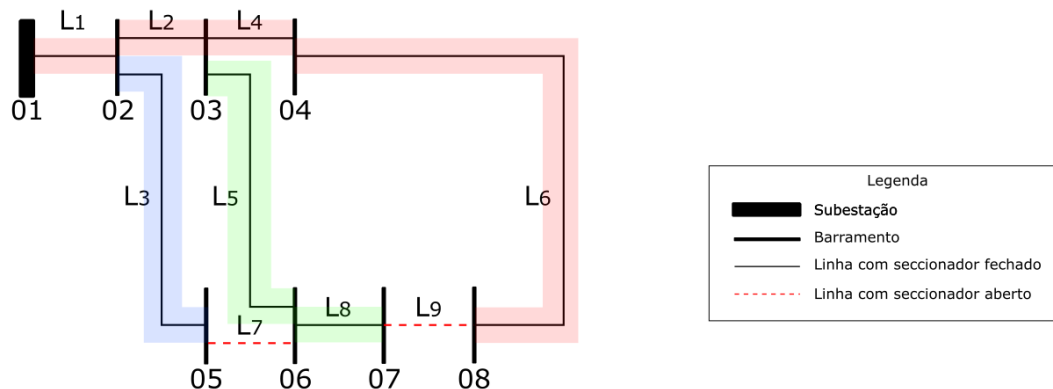


Figura 3.22: Mutação inteligente - 4ª configuração

No final, ao comparar todas as *fitness* obtidas para cada uma das configurações é escolhida a que tem menor valor, ou seja, a da Figura 3.22.

Na Figura 3.23 podemos ver uma representação gráfica da mutação inteligente. Enquanto a mutação simples se restringia a apenas um eixo, a mutação inteligente escolhe a melhor das opções apresentadas no novo eixo.

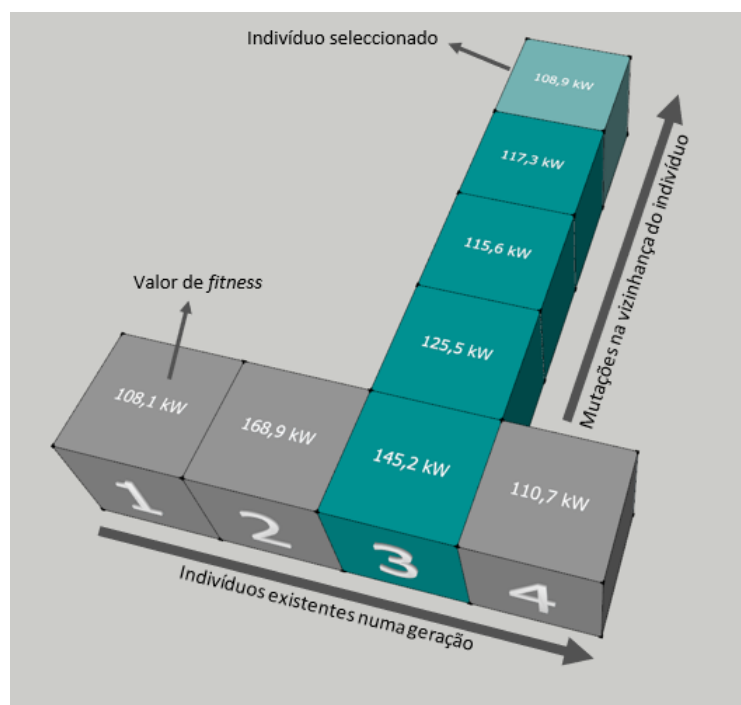


Figura 3.23: Esquema 3D que ilustra a mutação inteligente

3.3.4 Processo de selecção

Depois de realizar todas as operações genéticas a nova população excede evidentemente o número máximo de habitantes definido. Isto acontece porque todos os cruzamentos e

mutações simples e inteligentes são adicionadas à nova população.

Para diminuir este excesso de indivíduos é necessário eliminar os que têm maior valor de *fitness*, isto é, maior valor de perdas. A função criada com este propósito escolhe de todos os indivíduos aquele que tem maior valor de *fitness* e elimina essa solução. Este processo repete-se até que o número de indivíduos satisfaça o valor imposto pelo utilizador.

O código em *Python* respectivo ao desenvolvimento deste procedimento encontra-se no ponto 3 do anexo I.2.

3.3.5 Parametrização

O algoritmo genético desenvolvido varia de acordo com parametrização inicial que o utilizador faça. Neste problema, são parametrizáveis os seguintes valores:

- Número de indivíduos/membros da população N_{pop}
- Número de gerações N_{ger}
- Probabilidade de ocorrer mutação simples $Mut_{\%}$
- Probabilidade de ocorrer mutação inteligente $IntMut_{\%}$

Ao variar os parâmetros em vários testes foi possível concluir de que os valores ideais seriam:

- $N_{pop} = 7$, uma vez que com este número de membros é possível, através do cruzamento, criar no máximo 28 novos membros. Estes valores perante uma rede com o tamanho da utilizada para o estudo do capítulo 4 revelou-se ser suficiente.
- $N_{ger} = 10$, porque devido ao tamanho da rede *IEEE* apresentada em 4.2, este valor era mais do que suficiente para obter uma boa curva de evolução da *fitness* sem haver elevada saturação dos resultados.
- $Mut_{\%} = 30\%$
- $IntMut_{\%} = 20\%$

Uma vez que a mutação é o operador com maior impacto no processo evolutivo, o valor de ambos os parâmetros não deve ser muito elevado. No entanto, para a rede de 33 barramentos em estudo, são os valores que juntamente com os parâmetros anteriores permitem atingir bons resultados de forma rápida e eficaz.

O critério de fecho escolhido. neste caso, para o algoritmo genético, foi até atingir o número de gerações escolhidas no respectivo parâmetro(N_{ger}).

3.4 Ambiente de desenvolvimento

O *Software PSS/E* é uma ferramenta de simulação e análise de sistemas de energia eléctrica. Este *software*, desenvolvido pela *Siemens*, permite aos utilizadores executar uma ampla variedade de funções de análise que incluem trânsito de energia, análise dinâmica, cálculo de correntes curto-circuito, análise de contingências, calcular a estabilidade da tensão, entre outras. Utilizando as mais recentes tecnologias para automação de linguagem, o *PSS/E* introduziu o *Python* como nova linguagem de programação criando uma *Application Program Interface (API)*. Esta interface, está directamente ligada ao *Python*, fornecendo uma biblioteca com diversas funções que permitem realizar topo o tipo de operações que seriam possíveis no ambiente do próprio programa. O *Python* é uma linguagem de programação cujo objectivo é produzir código produtivo e legível de forma rápida. Assim, alguns das suas características principais são o reduzido uso de caracteres especiais e o recurso à indentação para definição de blocos. *Python* tem uma biblioteca padrão muito vasta, contendo métodos e funções que permitem realizar, essencialmente, qualquer tarefa. Esta linguagem tem liderado as tabelas nos últimos anos quando comparado a outras linguagens de programação como C, C++ e Java, e é amplamente utilizada por programadores. As empresas de desenvolvimento de software têm preferido o *Python* devido aos seus recursos versáteis e à necessidade de utilizar menos código na programação. Esta linguagem tem diversos tipos de aplicação, desde jogos, *frameworks*, aplicações *web*, desenvolvimento de linguagem, entre outros. Estas características fazem com que o *Python* seja uma linguagem com maior uso face a outras linguagens de programação utilizadas na indústria. Porém, como tudo, existem algumas das desvantagens atribuídas ao *Python* que passam pela fraca capacidade de computação em celulares, o processamento ocasionalmente lento, alguns erros de compilação e uma camada de acesso à base de dados subdesenvolvida e primitiva.

Foi com o objectivo de utilizar esta API que os algoritmos foram desenvolvidos em *Python*. Assim, é possível que operações como o trânsito de energia sejam calculadas directamente no *PSS/E*, verificando-se velocidades muito superiores a outras linguagens como, por exemplo, o *Matlab*.

O código desenvolvido em *Python* regista as reconfigurações topológicas e submete-as no *PSS/E*, onde este calcula o trânsito de energia e envia os resultados para o algoritmo. Em paralelo, o *Python* pode comunicar directamente com o *PSS/E* para utilizar as mais variadas funções. Sempre que oportuno, é possível guardar os dados que se pretendam em *Excel* através da biblioteca desenvolvida para *Python* chamada '*xlswriter*'. Outras bibliotecas como '*numpy*', '*random*' e '*pandas*' foram utilizadas para o desenvolvimento dos algoritmos.

APLICAÇÃO

4.1 Introdução

Nos capítulos anteriores foram propostos e apresentados dois métodos que permitem obter configurações para uma rede obtendo o menor valor de perdas possível, facilitando a localização preferencial de OCR's que possibilitam as reconfigurações topológicas necessárias a implementar, posteriormente, no terreno. Neste capítulo esses métodos serão aplicados a uma rede *IEEE* de 33 barramentos [17, 19] adaptada utilizada como caso de estudo à presente dissertação.

4.2 Caracterização da rede

A rede *IEEE* foi utilizada no artigo [19] como rede de teste do algoritmo de redução de perdas em redes de distribuição de energia eléctrica. Na dissertação [17], o método apresentado foi validado utilizando exactamente a mesma rede. Como tal, nesta dissertação, é possível, ao utilizar a rede *IEEE*, validar e ao mesmo tempo comparar os resultados dos métodos apresentados com os resultados obtidos em [17].

Na figura 4.1 é apresentada a estrutura da rede *IEEE*. Esta é constituída por uma subestação que alimenta uma carga total de 3,6 MW e 2,3 MVar, 33 barramentos e 37 linhas que fazem a ligação entre os diferentes barramentos. As linhas 12, 14, 18, 24 e 30 são linhas com seccionador aberto (N_{SA}) que permitem fazer a reconfiguração da rede. Os dados referentes às características das linhas encontram-se no anexo III.1.

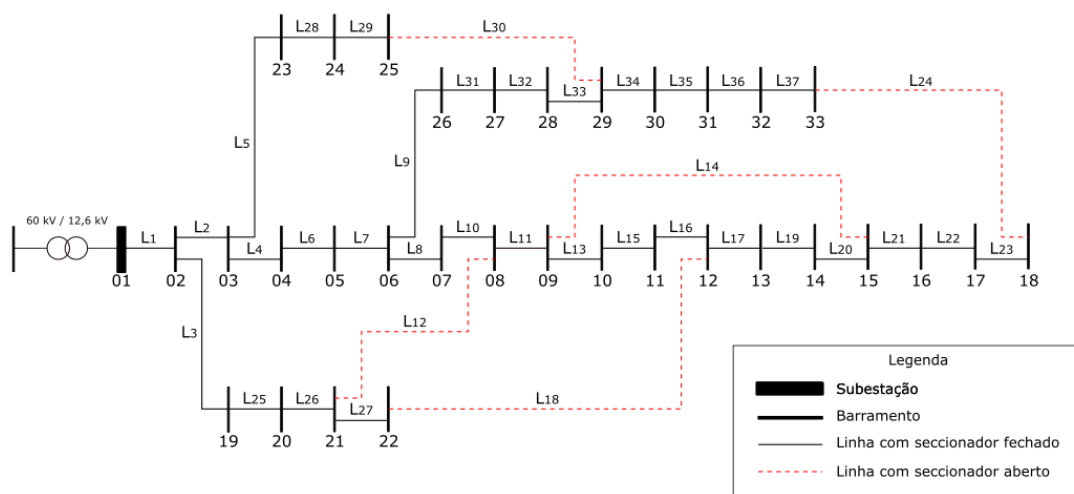


Figura 4.1: Rede IEEE de 33 barramentos

Para efeitos de fiabilidade foi testado, para cada um dos algoritmos, três cenários diferentes - carga normal, aumento (+20%) de carga e redução (-20%) de carga -, de forma a observar como podem variar as configurações da rede perante a variação de carga. Assim, é possível fazer uma opção mais acertada na escolha do local ideal de colocação dos OCR's. Os valores de carga utilizados encontram-se detalhadamente apresentados no anexo III.2.

4.3 Aplicação das diferentes metodologias à rede apresentada

4.3.1 Cenário: Base

A configuração inicial da rede IEEE está apresentada na figura 4.2.

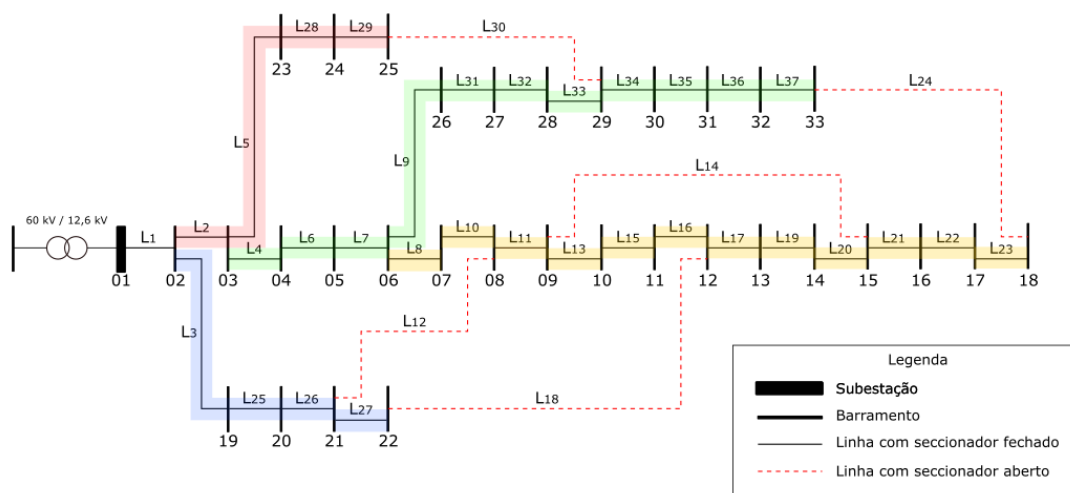


Figura 4.2: Configuração inicial da rede - Cenário: Base

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

4.3.1.1 Resultados Método de busca sequencial

Depois de aplicado o método de busca sequencial, exemplificado no capítulo 3.2, obtém-se a seguinte configuração final:

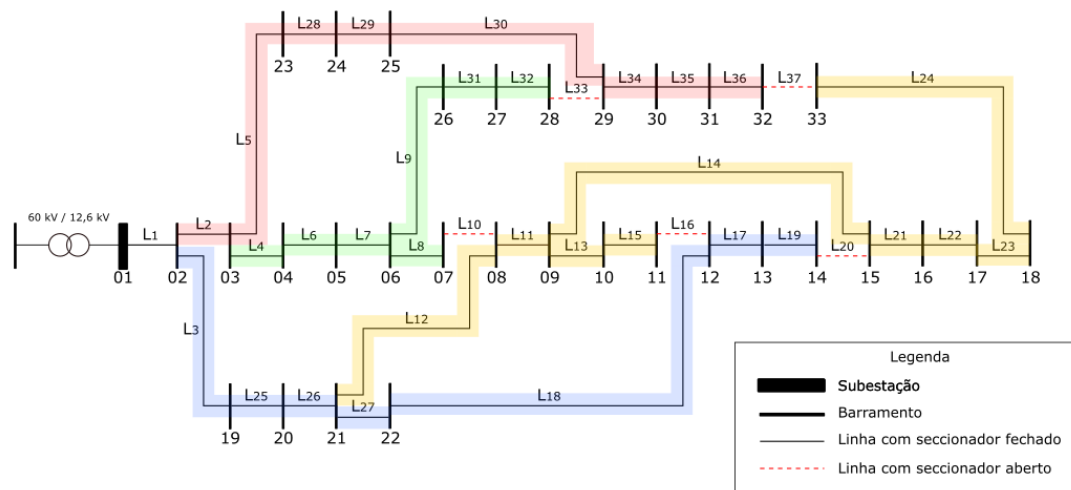


Figura 4.3: Configuração final da rede - Cenário: Base

Alterando a configuração inicial da rede, foi possível verificar que ligando as linhas 12, 14, 18, 24 e 30, e desligando as linhas 10, 16, 20, 33 e 37, é possível reduzir as perdas de 172,5 kW para 110,1 kW, representando uma redução de 62,4 kW, ou seja, na ordem dos 36%. Os resultados de todas as iterações encontram-se no anexo V.1.

Pela análise do gráfico da figura 4.4 é possível observar que a diferença do valor das tensões nodais entre barramentos diminuiu significativamente. De referir que, quanto menor for a diferença de tensão entre dois barramentos menor serão as perdas por efeito de *joule*, uma vez que o trânsito de energia é reduzido.

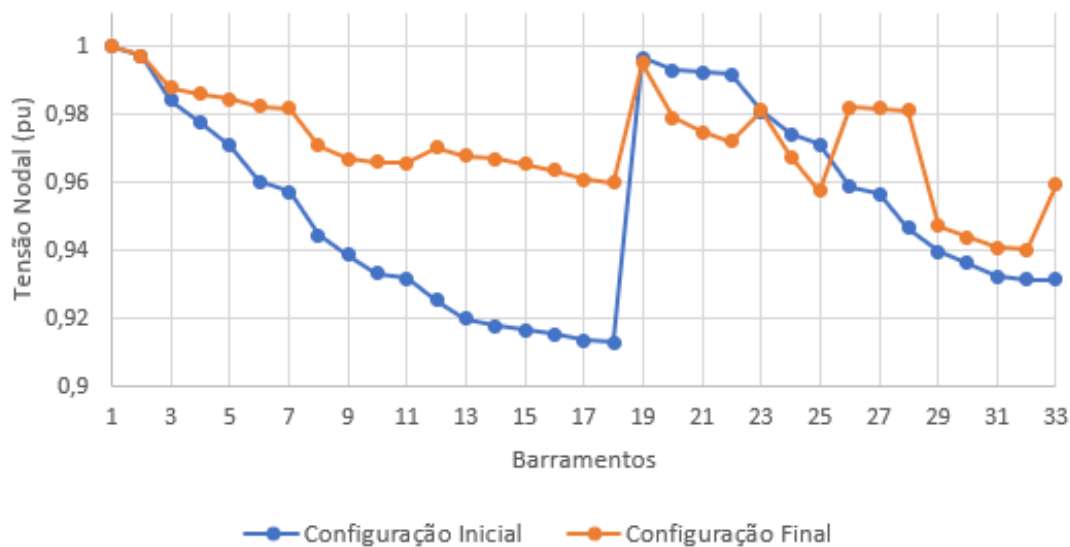


Figura 4.4: Gráfico comparativo das tensões nodais - Cenário: Base

Na configuração da figura 4.5, o autor da dissertação [17] reduziu as perdas de 197,8kW para 130,0kW, o que corresponde a uma redução de 34%.

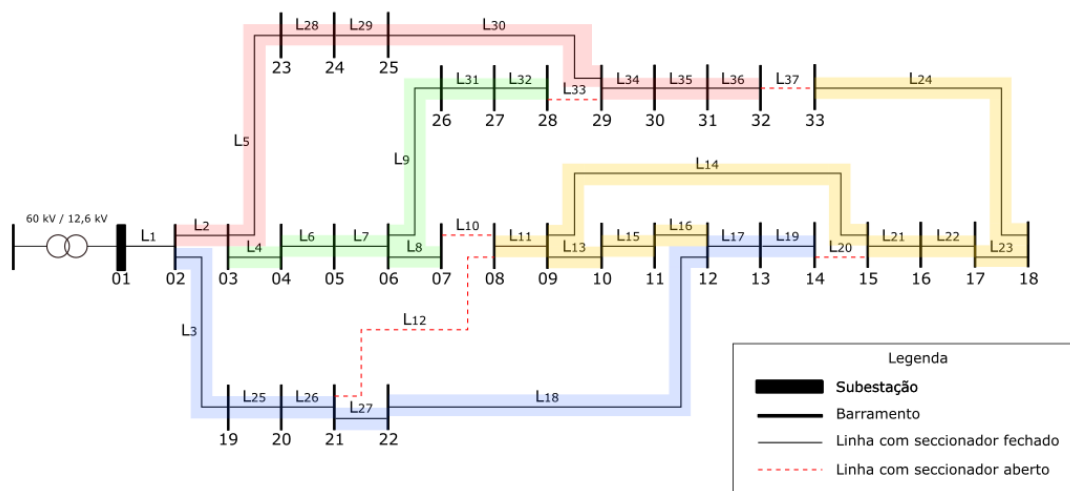


Figura 4.5: Configuração final na dissertação [17] - Cenário: Base

No gráfico da figura 4.6 observa-se que os valores das tensões nodais são bastante próximos uns dos outros, o que seria de esperar uma vez que a diferença de perdas determinada pelos dois estudos é de 2%.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

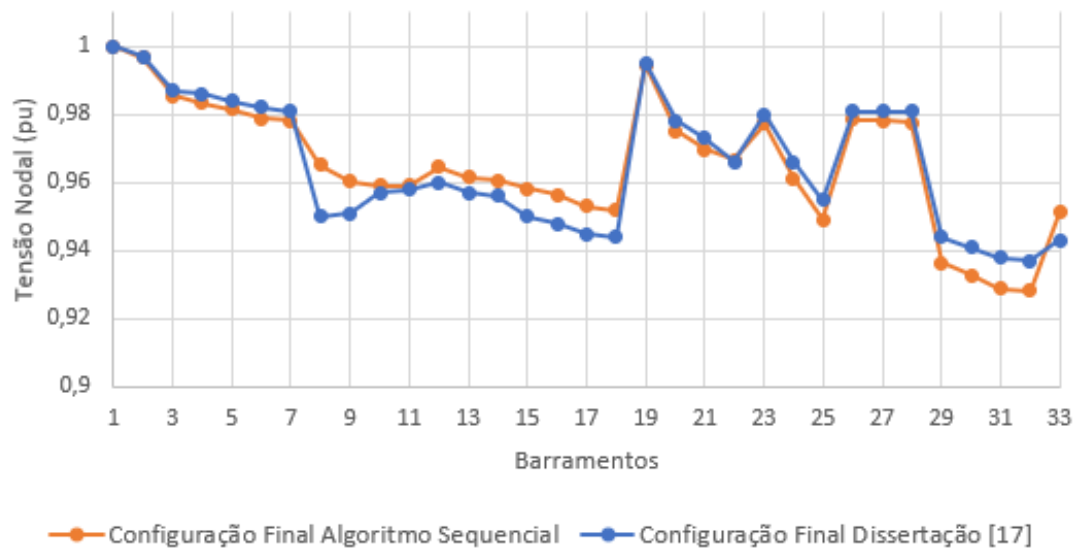


Figura 4.6: Comparação das tensões nodais entre a configuração apresentada e a dissertação [17] - Cenário: Base

Todavia, uma vez que os valores são próximos e como o algoritmo da dissertação [17] foi validado comparativamente ao do artigo [19], é possível fazer a validação do método de busca sequencial apresentado relativamente a ambos os autores.

4.3.1.2 Resultados Algoritmo Genético

Para o algoritmo genético, a parametrização dada no sub-capítulo 3.3.5 foi a utilizada ao longo de todas as simulações e foram sempre guardadas as 10 melhores topologias encontradas ao longo das várias gerações. Contudo, serão apenas apresentadas as configurações das 3 topologias com menor valor de perdas.

Antes de se apresentar as configurações, na figura 4.7 mostra-se a evolução da *fitness* ao longo das várias gerações.

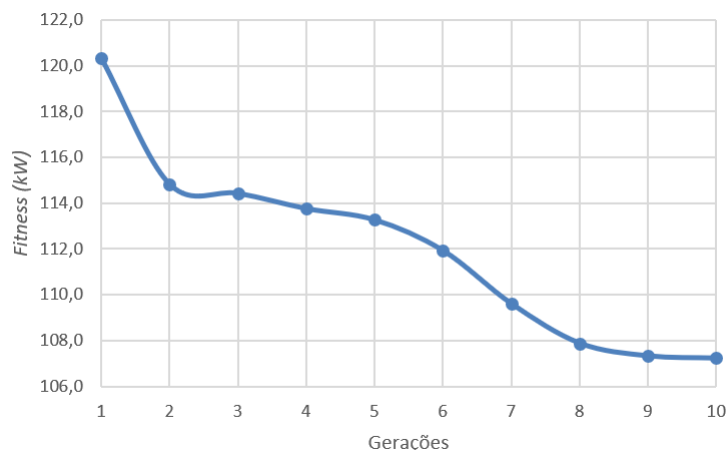


Figura 4.7: Evolução da função *fitness* ao longo das várias gerações - Cenário: Base

Esta evolução vai ao encontro do expectável para um bom algoritmo genético, uma vez que a redução é progressiva ao longo das gerações.

A configuração inicial é a mesma da figura 4.2.

Na configuração da figura 4.8 o valor de perdas foi reduzido de 172,5kW para 107,2kW, o que representa uma redução de 38%.

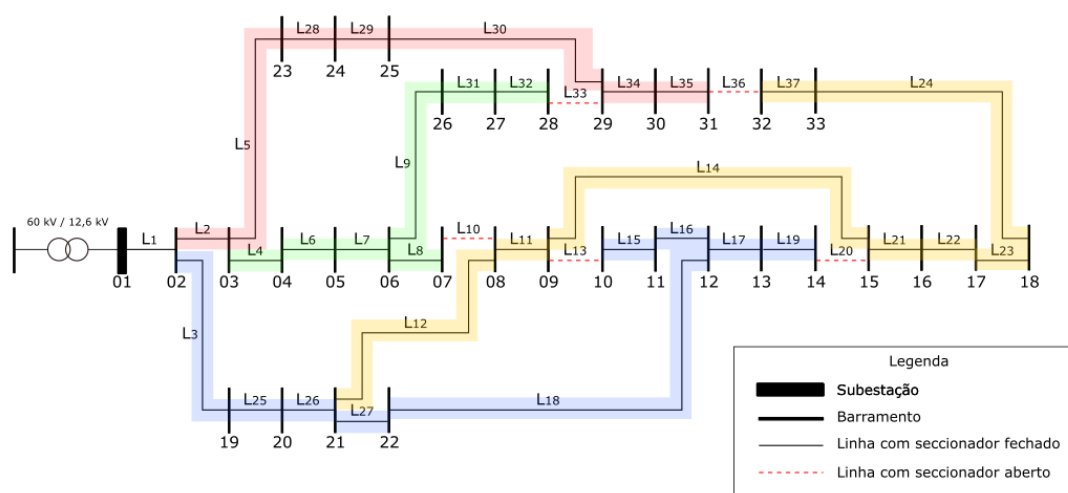


Figura 4.8: 1ª configuração final (AG) - Cenário: Base

Na configuração da figura 4.9 o valor de perdas é de 107,5kW.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

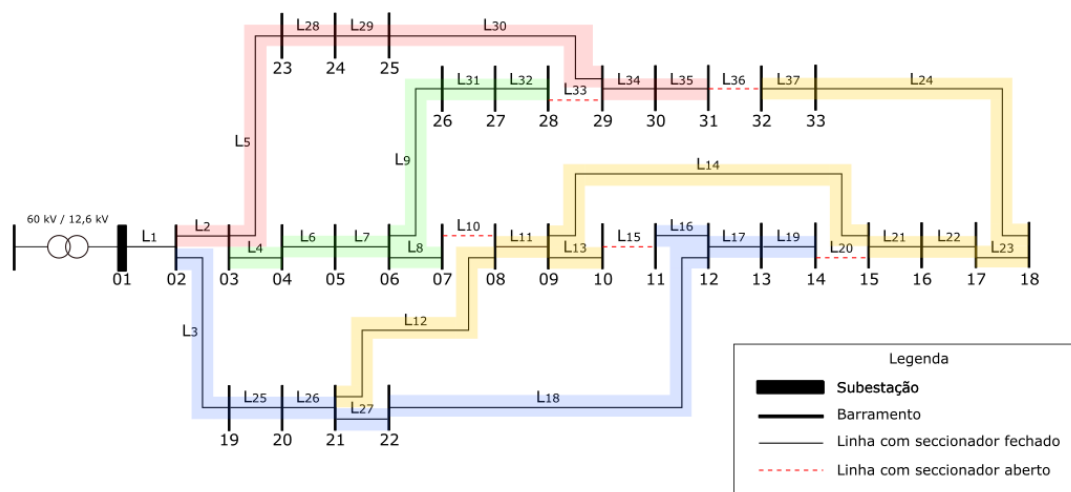


Figura 4.9: 2ª configuração final (AG) - Cenário: Base

Na configuração da figura 4.10 o valor de perdas é de 107,9kW.

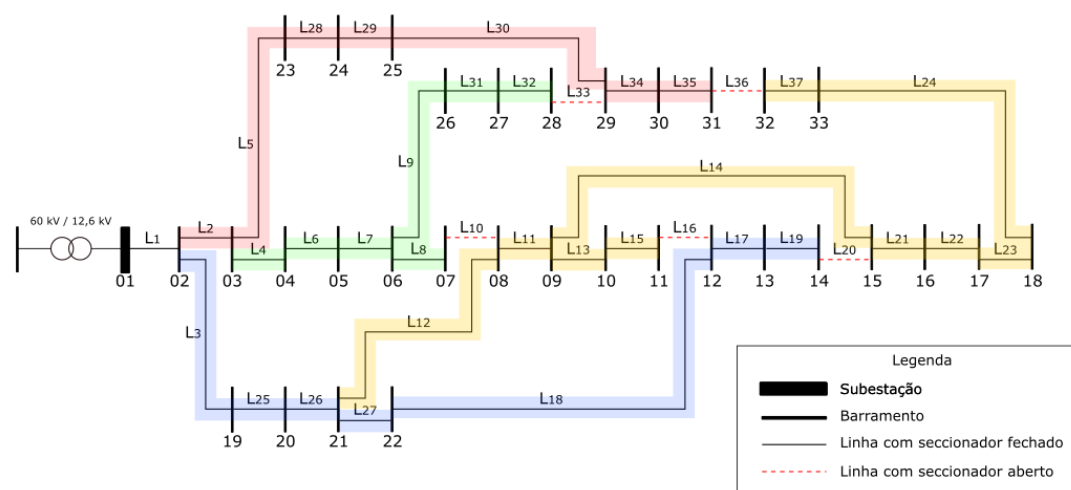


Figura 4.10: 3ª configuração final (AG) - Cenário: Base

O gráfico da figura 4.11 ilustra a diferença de tensão entre os barramentos na 1ª configuração final (Figura 4.8.

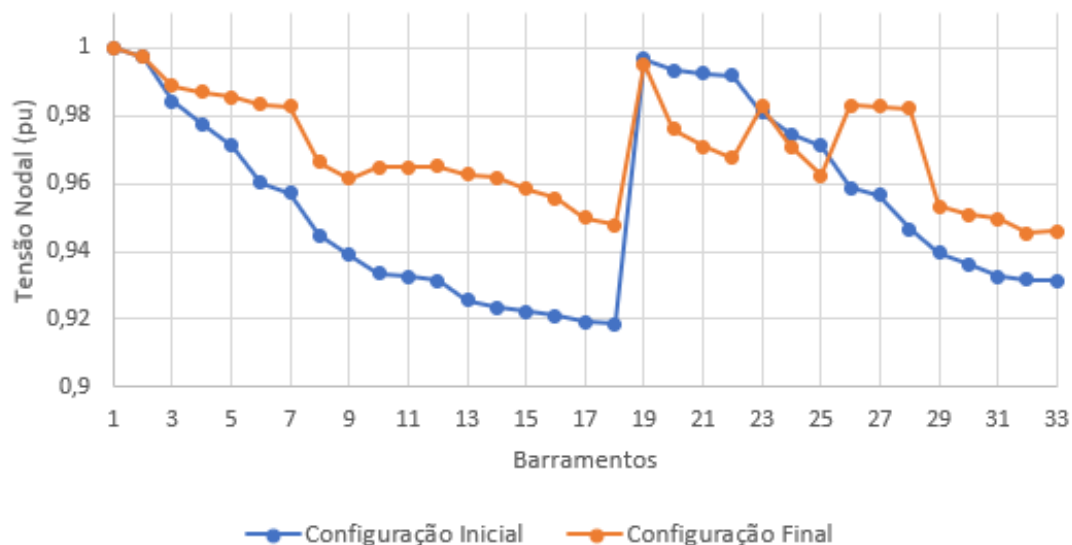


Figura 4.11: Gráfico comparativo das tensões nodais (AG) - Cenário: Base

A tabela da figura 4.1 mostra as 10 melhores *fitness* que foram obtidas para o cenário em análise.

Cenário: Base		
Configuração	Perdas (kW)	
Inicial	172,5	
Final Iterativo	110,1	
<i>fitness</i> AG - 1	107,2	Fig 4.8
<i>fitness</i> AG - 2	107,5	Fig 4.9
<i>fitness</i> AG - 3	107,9	Fig 4.10
<i>fitness</i> AG - 4	108,2	Anexo VI.1
<i>fitness</i> AG - 5	108,8	
<i>fitness</i> AG - 6	109,4	
<i>fitness</i> AG - 7	112,2	
<i>fitness</i> AG - 8	112,4	
<i>fitness</i> AG - 9	112,4	
<i>fitness</i> AG - 10	113,2	

Tabela 4.1: Tabela com os melhores valores de *fitness* - Cenário: Base

Comparativamente ao método de busca sequencial, foi possível não só reduzir mais 2% o valor total de perdas activas, como obter várias topologias semelhantes com valores bastante próximos. Na figura 4.12 é possível observar a comparação entre as tensões nodais da melhor configuração do algoritmo genético com as tensões nodais do método de busca sequencial.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

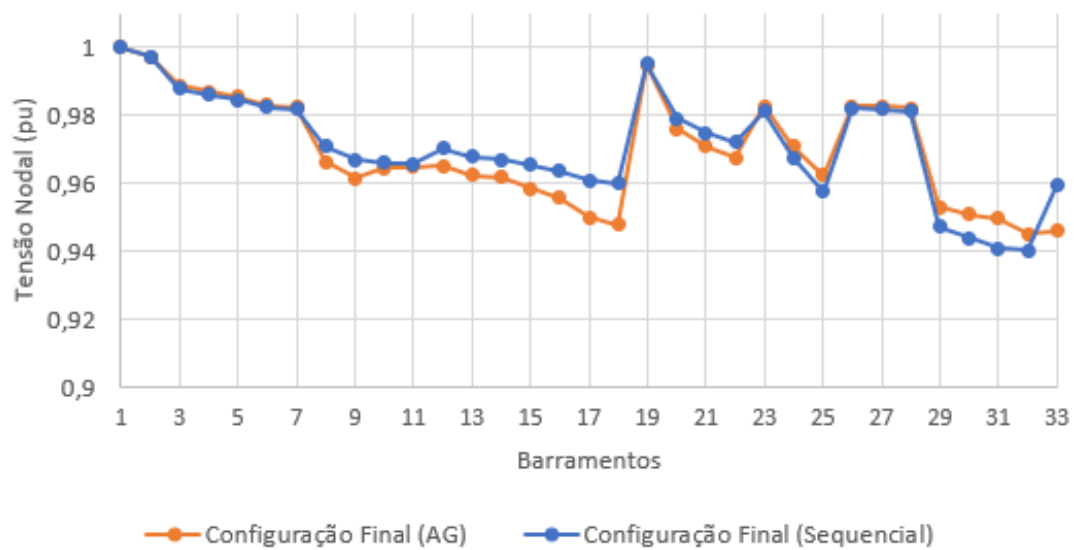


Figura 4.12: Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Base

Através da análise das diferentes configurações apresentadas, é possível concluir que com, apenas algumas décimas de diferença entre o valor de perdas, é possível obter diferentes configurações para a mesma rede, o que permite uma maior diversidade na escolha da localização preferencial dos OCR's.

4.3.2 Cenário: Redução de carga

A configuração inicial da rede *IEEE* para o cenário de redução de carga está apresentada na figura 4.13.

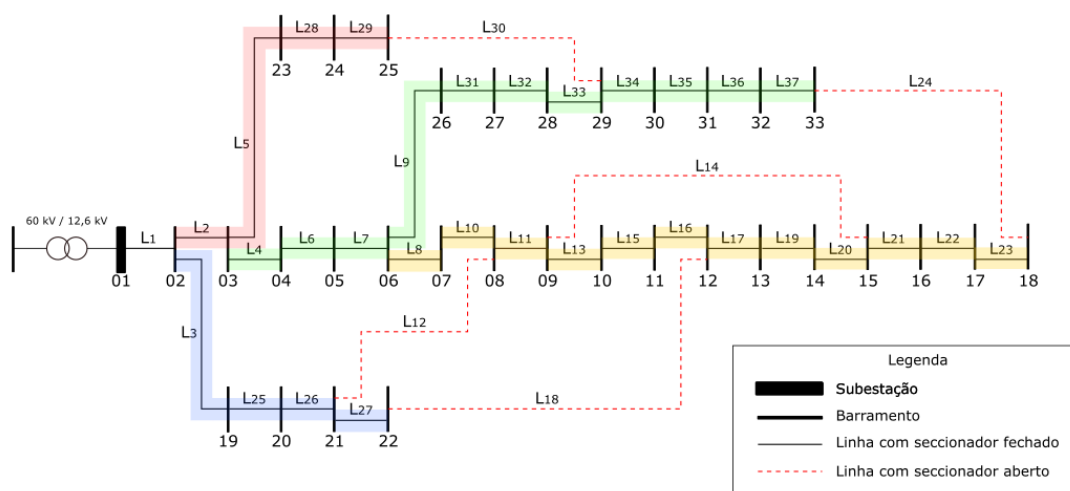


Figura 4.13: Configuração inicial da rede - Cenário: Redução

4.3.2.1 Resultados Método de busca sequencial

Aplicando o método de busca sequencial obtém-se a seguinte configuração final:

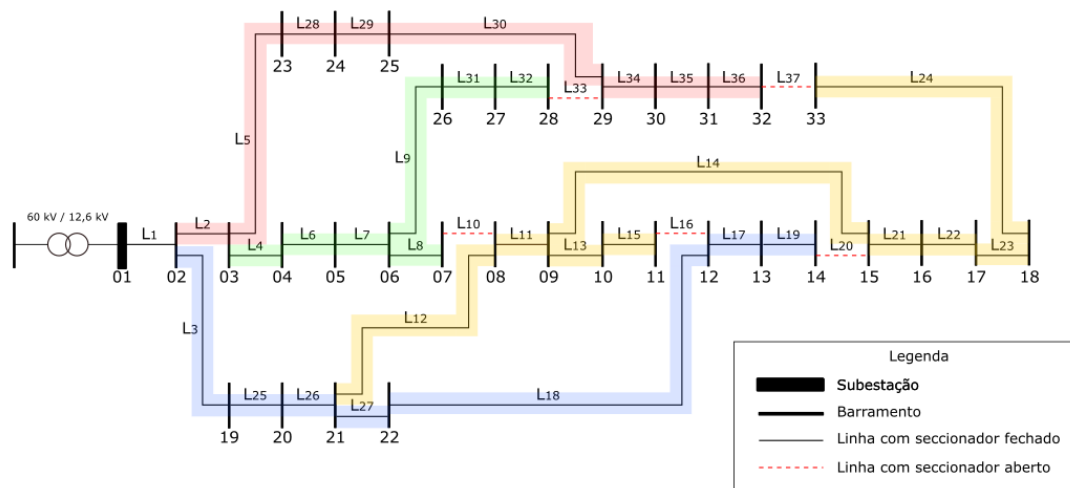


Figura 4.14: Configuração final da rede - Cenário: Redução de carga

Pela análise da figura, observa-se, *a priori*, que a configuração é exactamente igual à do cenário base apresentado no capítulo 4.3.1.1. Neste cenário as perdas foram reduzidas de 110,5kW para 70,5kW, ou seja, na ordem dos 36%, igual ao cenário base, o que seria de esperar, uma vez que a configuração é exactamente a mesma. Os resultados de todas as iterações encontram-se no anexo V.2.

Porém, e uma vez que a carga de cada barramento foi reduzida 20% relativamente ao cenário base, é possível verificar através da figura 4.15 que os valores das tensões nodais diminuíram significativamente tanto na configuração inicial como na final.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

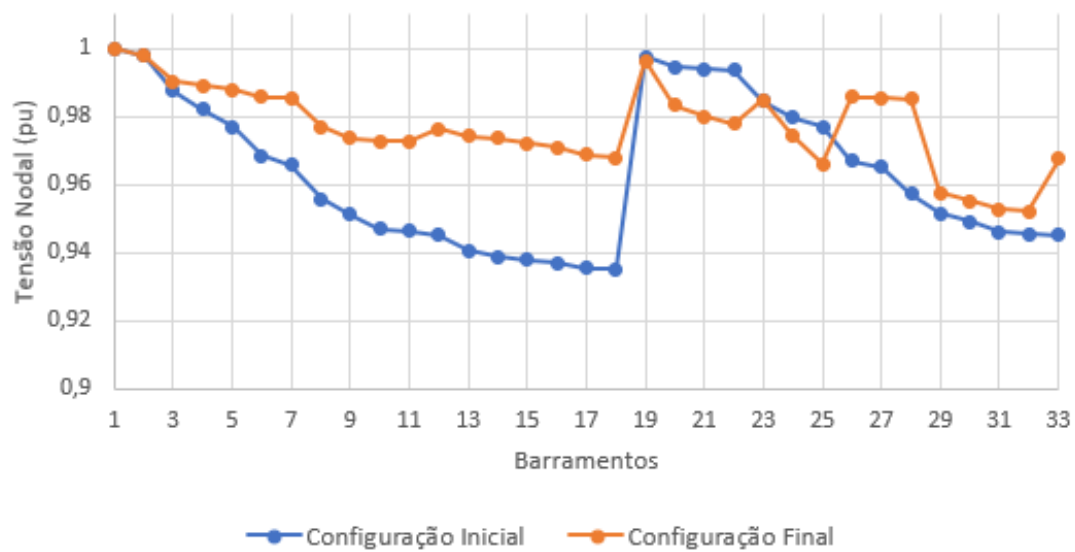


Figura 4.15: Gráfico comparativo das tensões nodais - Cenário: Redução de carga

4.3.2.2 Resultados Algoritmo Genético

Tal como no cenário anterior, foi aplicado o algoritmo genético à rede *IEEE* apresentada.

O gráfico da figura 4.16 permite observar a evolução da *fitness* ao longo das dez gerações.

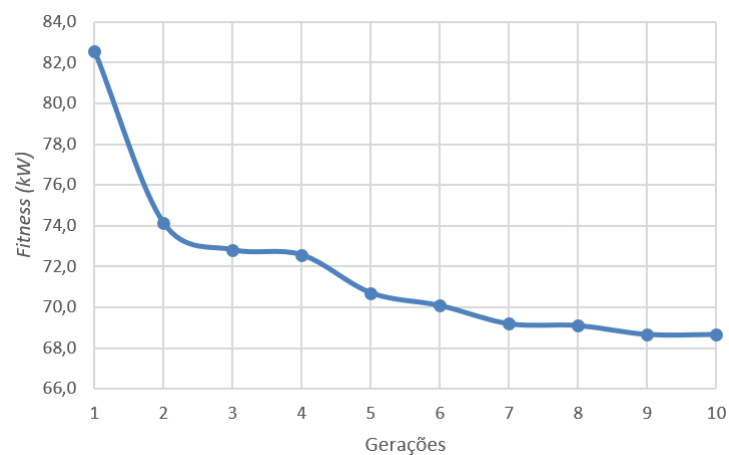
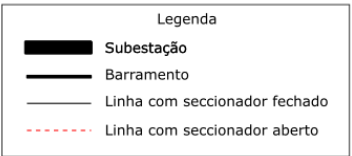
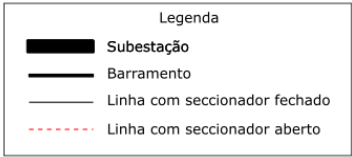


Figura 4.16: Evolução da função *fitness* ao longo das gerações - Cenário: Redução de carga

Novamente, a evolução da *fitness* mantém-se dentro do expectável apresentado uma redução constante. A configuração inicial neste cenário é ilustrada pela figura 4.13.

Na configuração da figura 4.17 o valor de perdas foi reduzido de 110,5kW para 68,7kW, o que representa uma redução de 38%.



4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

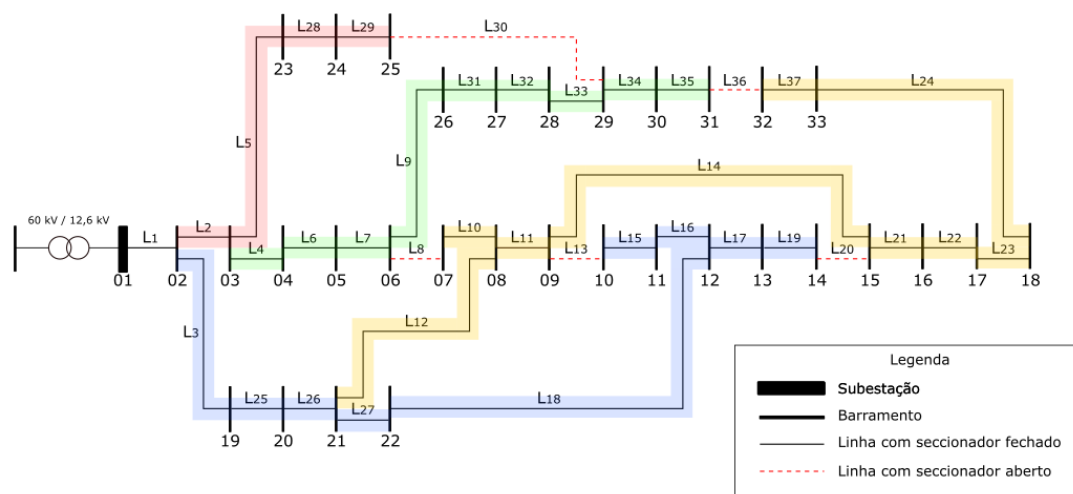


Figura 4.19: 3ª configuração final (AG) - Cenário: Redução de Carga

O gráfico da figura 4.20 ilustra a diferença de tensão entre os barramentos na 1ª configuração final (Figura 4.17).

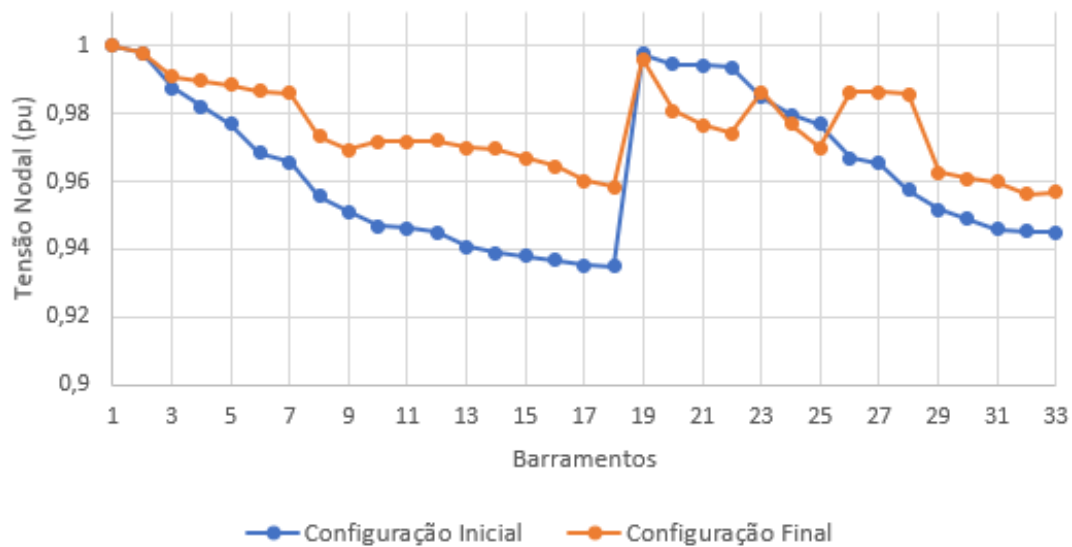


Figura 4.20: Gráfico comparativo das tensões nodais (AG) - Cenário: Redução de carga

A tabela da figura 4.2 mostra as 10 melhores *fitness* que foram obtidas neste cenário.

Cenário: Redução de carga		
Configuração	Perdas (kW)	
Inicial	110,5	
Final Iterativo	70,5	
<i>fitness</i> AG - 1	68,7	Fig 4.18
<i>fitness</i> AG - 2	69,2	Fig 4.19
<i>fitness</i> AG - 3	70,5	Fig 4.20
<i>fitness</i> AG - 4	70,8	Anexo VI.2
<i>fitness</i> AG - 5	72,1	
<i>fitness</i> AG - 6	72,9	
<i>fitness</i> AG - 7	73,0	
<i>fitness</i> AG - 8	73,5	
<i>fitness</i> AG - 9	75,1	
<i>fitness</i> AG - 10	75,1	

Tabela 4.2: Tabela com os melhores valores de *fitness* - Cenário: Redução de carga

Na figura 4.21 é possível observar a comparação entre as tensões nodais da melhor configuração do algoritmo genético com as tensões nodais do método de busca sequencial.

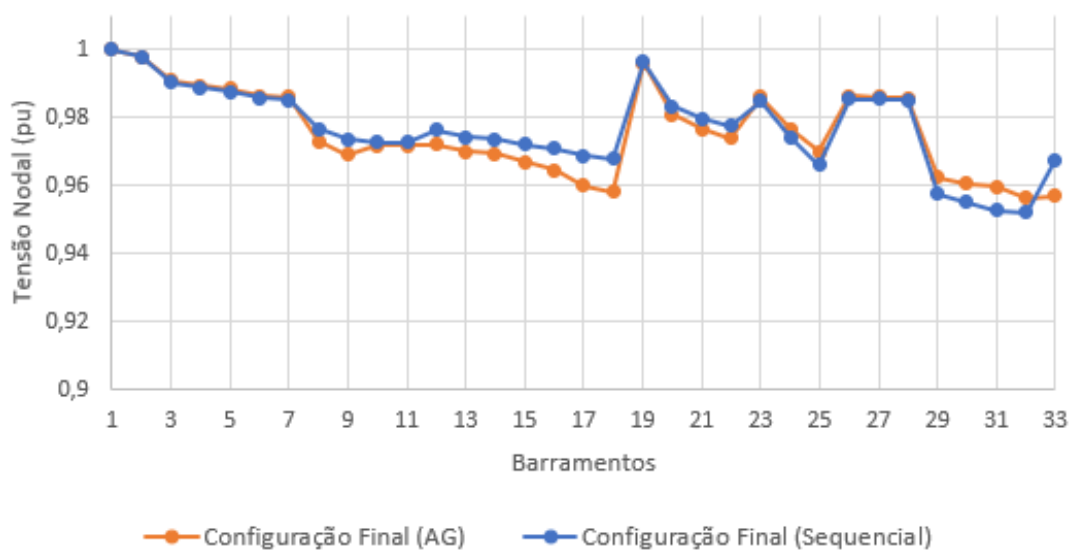


Figura 4.21: Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Redução de carga

4.3.3 Cenário: Aumento de carga

A configuração inicial da rede *IEEE* para o cenário de redução de carga está apresentada na figura 4.22.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

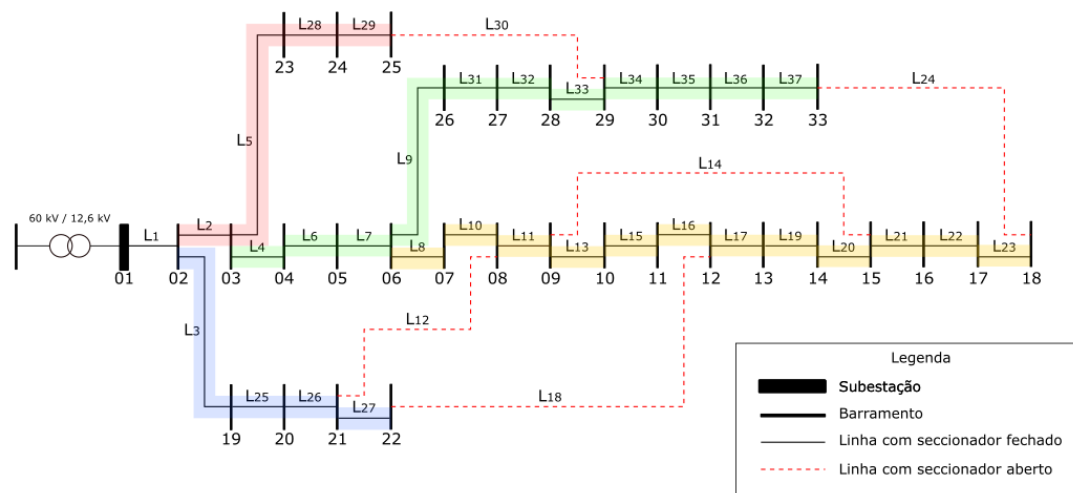


Figura 4.22: Configuração inicial da rede - Cenário: Aumento de carga

4.3.3.1 Resultados Método de busca sequencial

Como se pode observar na figura 4.23, aplicando ao cenário de aumento de carga o método de busca sequencial obtém-se a mesma configuração apresentada nos cenários anteriores:

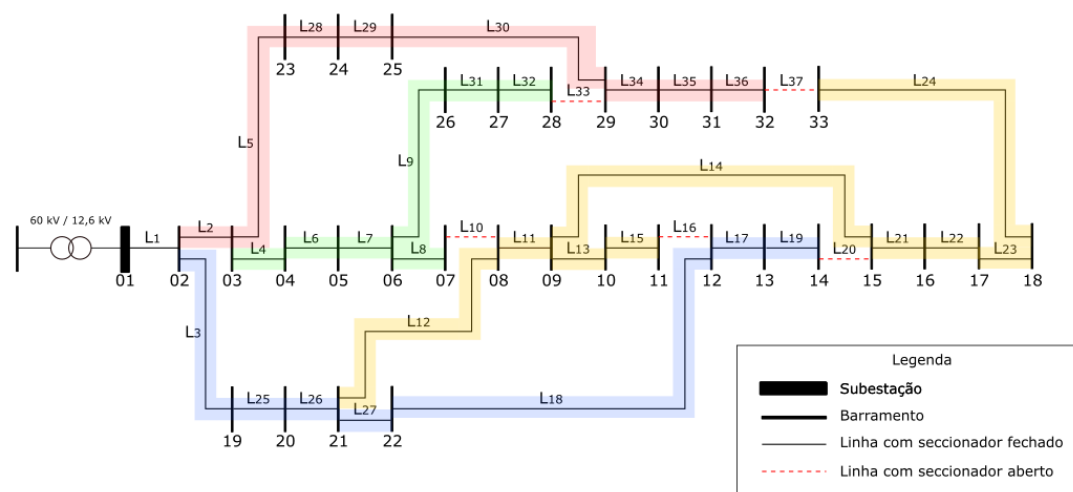


Figura 4.23: Configuração final da rede - Cenário: Aumento de carga

Neste cenário as perdas foram reduzidas de 248,1kW para 158,5kW, ou seja, na ordem dos 36%, mantendo a mesma percentagem. Os resultados de todas as iterações encontram-se no anexo V.3.

Da mesma forma que no cenário de redução de carga, as tensões nodais diminuíram, neste cenário os valores nodais das tensões subiram, como é possível verificar na figura 4.24:

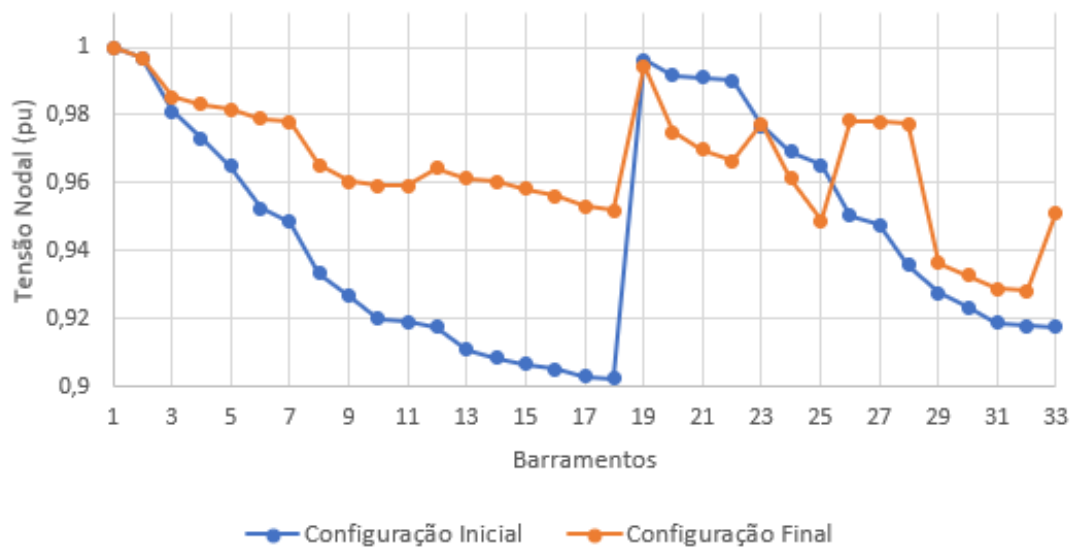


Figura 4.24: Gráfico comparativo das tensões nodais - Cenário: Aumento de carga

4.3.3.2 Resultados Algoritmo Genético

O gráfico da figura 4.25 permite observar a evolução da *fitness* ao longo das dez gerações.

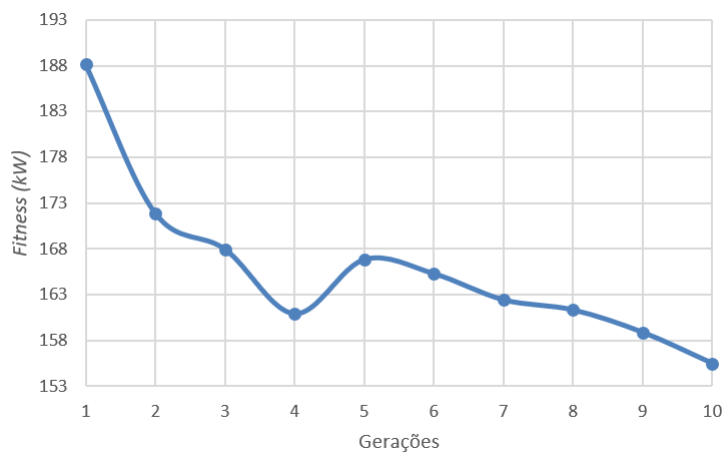


Figura 4.25: Evolução da função *fitness* ao longo das gerações - Cenário: Aumento de carga

Neste cenário, embora tenha havido uma 'deslocação' do valor médio da *fitness* na 4 geração em relação às seguintes gerações, a função *fitness* é boa. Contudo, é possível, através da análise da sua evolução, que tenha havido uma configuração que se destacou claramente das outras mas que pode não ter sobrevivido ao longo das restantes gerações, uma vez que, as soluções variam com os operadores genéticos e há diversas formas de implementar essas operações.

A configuração inicial para este cenário encontra-se na figura 4.22. Na configuração da figura 4.26 o valor de perdas foi reduzido de 248,1kW para 131,3kW, o que representa

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

uma redução de 47%.

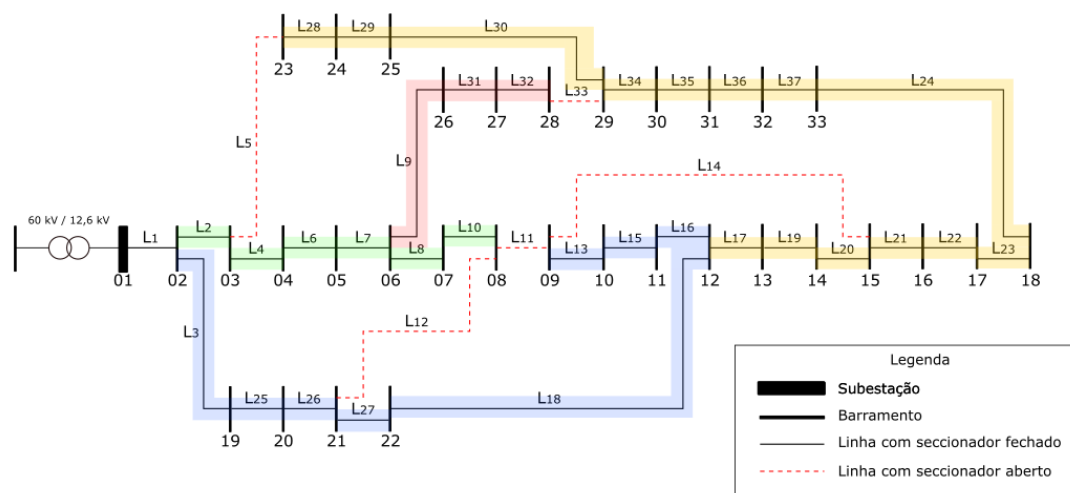


Figura 4.26: 1ª configuração final (AG) - Cenário: Aumento de Carga

Na configuração da figura 4.27 o valor de perdas é de 154,3kW.

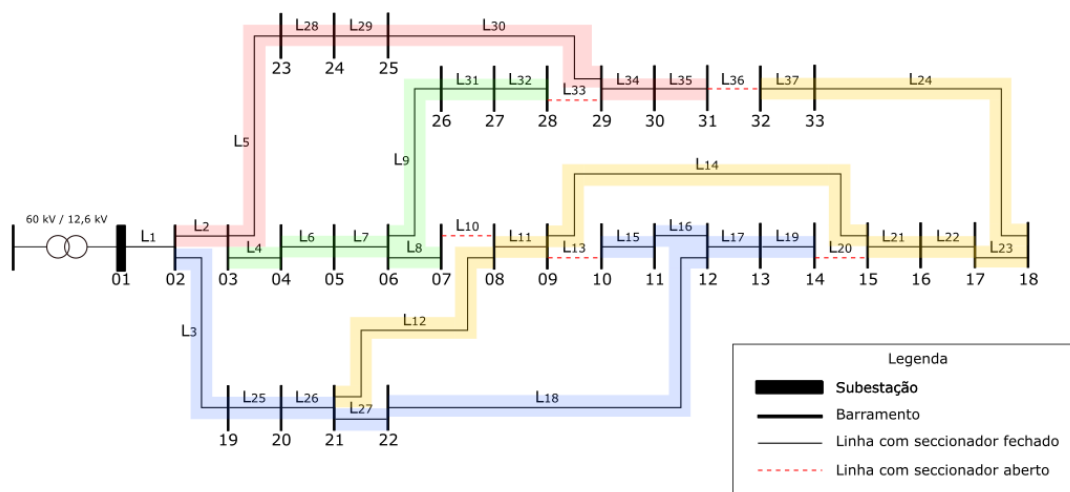


Figura 4.27: 2ª configuração final (AG) - Cenário: Aumento de Carga

Na configuração da figura 4.28 o valor de perdas é de 155,3kW.

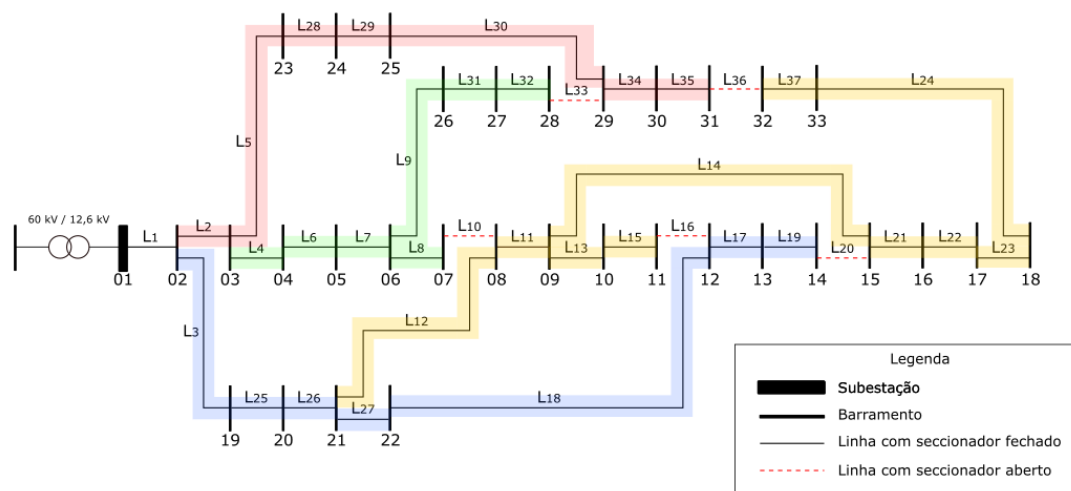


Figura 4.28: 3ª configuração final (AG) - Cenário: Aumento de Carga

O gráfico da figura 4.29 ilustra a diferença de tensão entre os barramentos na 1ª configuração final (Figura 4.26).

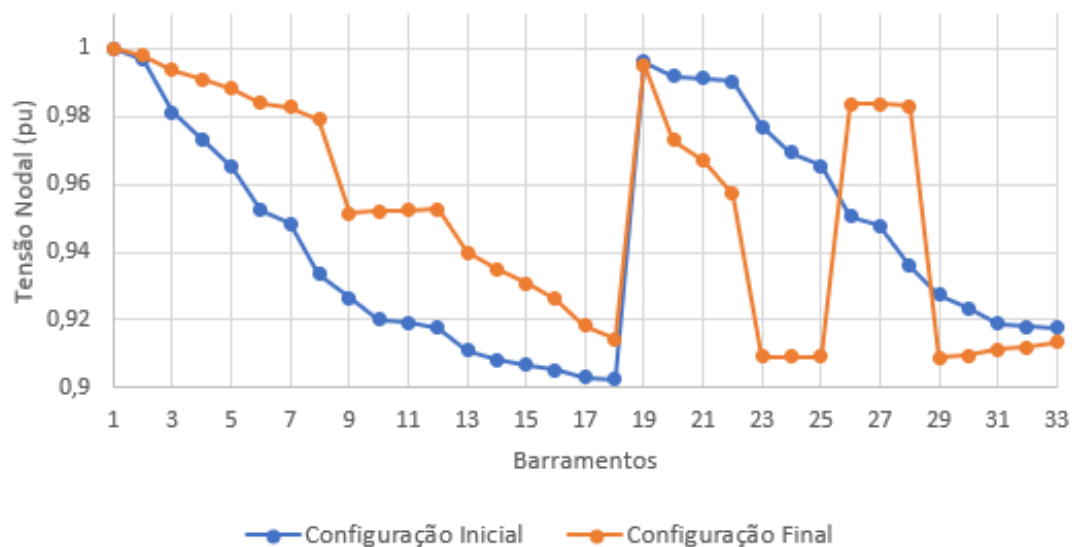


Figura 4.29: Gráfico comparativo das tensões nodais (AG) - Cenário: Aumento de carga

A tabela da figura 4.3 ilustra as 10 melhores *fitness* que foram obtidas para o cenário em análise.

4.3. APLICAÇÃO DAS DIFERENTES METODOLOGIAS À REDE APRESENTADA

Cenário: Aumento de carga		
Configuração	Perdas (kW)	
Inicial	248,1	
Final Iterativo	158,5	
<i>fitness</i> AG - 1	131,3	Fig 4.28
<i>fitness</i> AG - 2	154,3	Fig 4.29
<i>fitness</i> AG - 3	155,3	Fig 4.30
<i>fitness</i> AG - 4	156,3	Anexo VI.3
<i>fitness</i> AG - 5	157,4	
<i>fitness</i> AG - 6	158,5	
<i>fitness</i> AG - 7	160,9	
<i>fitness</i> AG - 8	162,1	
<i>fitness</i> AG - 9	163,0	
<i>fitness</i> AG - 10	163,7	

Tabela 4.3: Tabela com os melhores valores de *fitness* - Cenário: Aumento de carga

Tal como se esperava, é possível verificar que a 1ª configuração final apresentada na figura 4.26 é significativamente melhor do que as restantes 9 configurações, apresentando uma diferença de 23kW para a 2ª configuração e 32,4kW para a 10ª configuração, como se pode observar na tabela anterior.

Na figura 4.30 é possível observar a comparação entre as tensões nodais da melhor configuração do algoritmo genético com as tensões nodais do método de busca sequencial para o mesmo cenário.

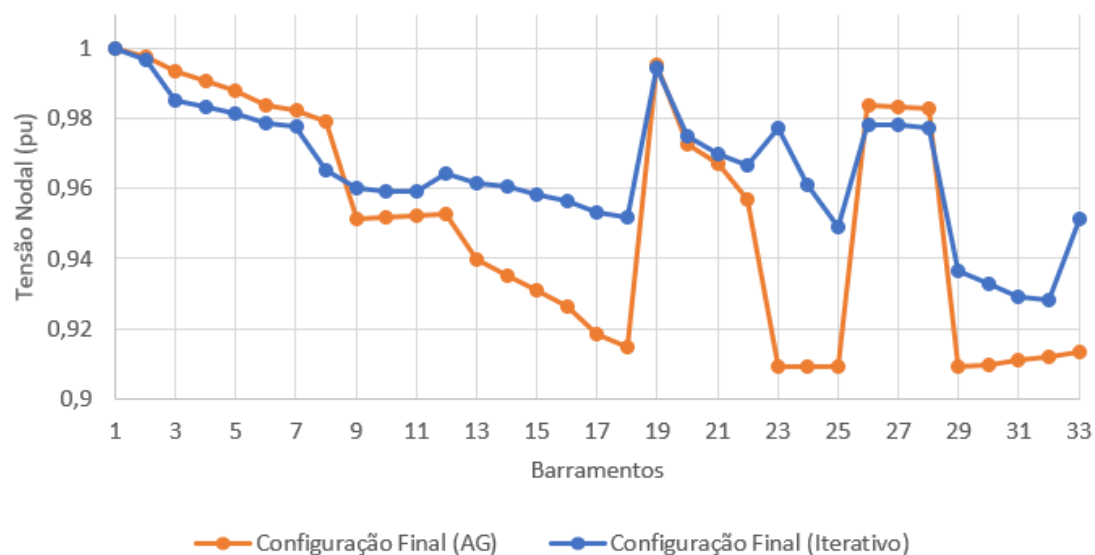


Figura 4.30: Comparação das tensões nodais entre o algoritmo genético e de busca sequencial - Cenário: Aumento de carga

4.4 Proposta de localização preferencial de OCR's

Antes de passar à definição subsequente das localizações dos OCR's, é importante referir que nesse processo apenas foram considerados os resultados do Algoritmo Genético. Isto porque, como foi possível observar ao longo do capítulo 4.3, analisando todos os resultados de ambos os métodos, ficou claro de que mesmo com variações de carga ao longos dos diferentes cenários, a configuração final com menos perdas activas dada pelo método sequencial era sempre igual. Por sua vez, tal não se verificou no AG onde foi possível obter várias configurações, todas elas diferentes entre si para cada cenário mas com valores de perdas activas bastante próximas.

Assim, analisando todos os resultados obtidos através do AG é possível verificar quais as linhas com seccionador aberto se repetiram mais vezes. Cada configuração tem 5 linhas com SA e para cada cenário obteve-se 10 configurações diferentes, que se encontram aglomeradas em 150 manobras. De forma a visualizar todas as manobras, foi desenvolvido o *heatmap* que se encontra na Figura 4.31. Nesta figura é possível verificar quais as linhas com SA que foram registadas ao longo das diferentes configurações e o número de vezes em que as mesmas foram registadas.

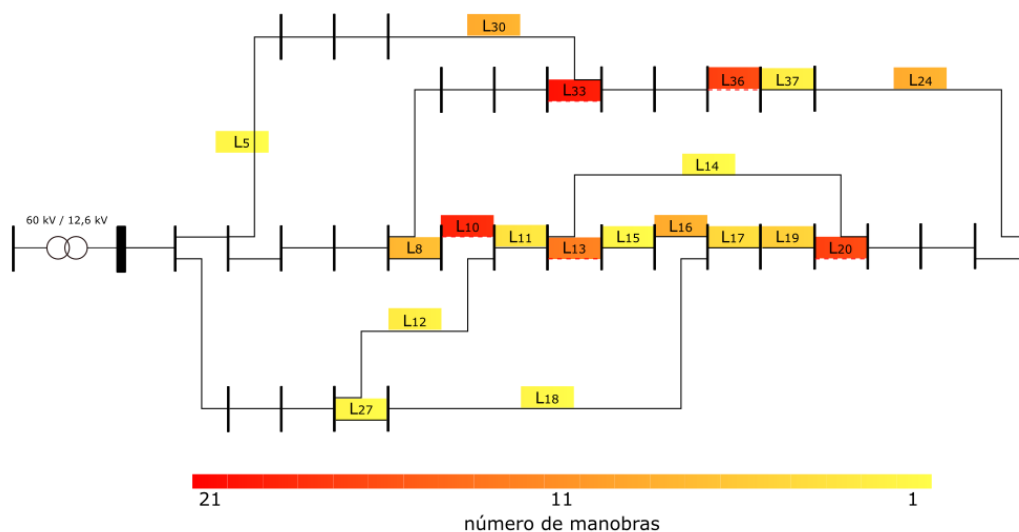


Figura 4.31: *Heatmap* que contém o número de manobras verificadas nos 3 cenários

Através da Figura 4.31 é fácil observar que as linhas abertas onde se verificaram mais manobras foram as linhas 10, 13, 20, 33 e 36. Assim, conclui-se de que estas linhas se devem manter abertas para cada um dos cenários apresentados.

Contudo, antes de passar ao objectivo final desta dissertação, isto é, a localização preferencial de OCR's que possibilitem por reconfiguração topológica a redução de perdas de energia numa rede de distribuição, é importante olhar, também, para os respectivos *heatmaps* de cada um dos cenários.

Na figura 4.32, encontra-se representado o *heatmap* com as manobras mais observadas

nas 10 configurações do Cenário: Base.

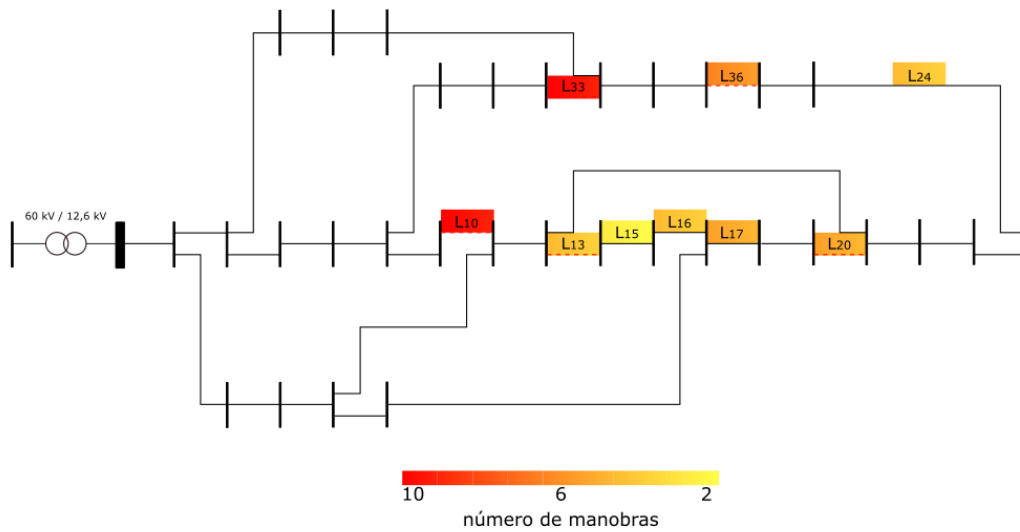


Figura 4.32: *Heatmap* que contém o número de manobras verificadas nos Cenário: Base

No *heatmap* do Cenário: Base, verificamos que para além das linhas 10 e 33, que são pontos destacados, também se verificam algumas manobras nas linhas 16, 17 e 20.

Por sua vez, as manobras verificadas ao longo das 10 configurações do Cenário: Redução de carga, estão representadas na figura 4.33

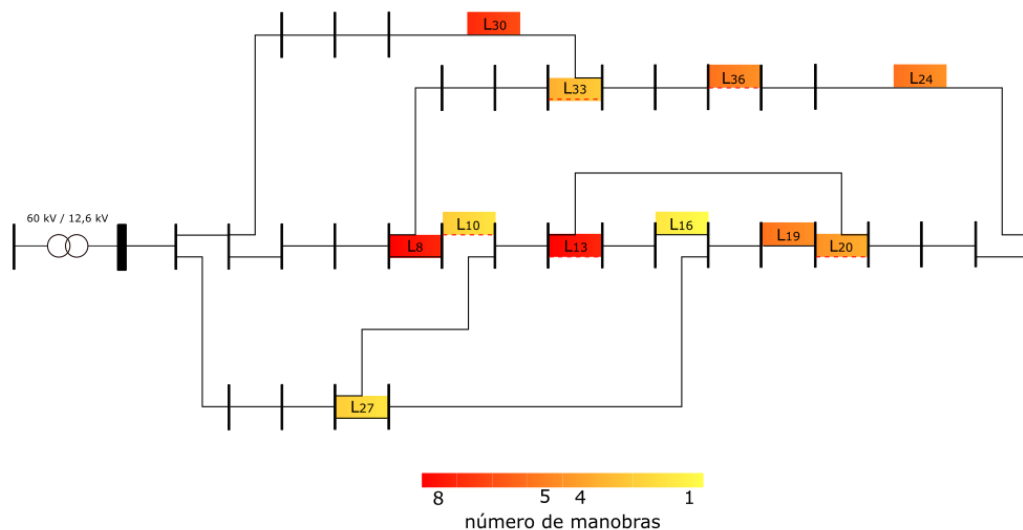


Figura 4.33: *Heatmap* que contém o número de manobras verificadas nos Cenário: Redução de carga

Neste cenário, verificou-se maior incidência de manobras nas linhas 8, 13 e 30.

Já nas configurações do Cenário: Aumento de carga, o *heatmap* de manobras também apresenta diversidade de resultados como se pode observar na figura 4.34

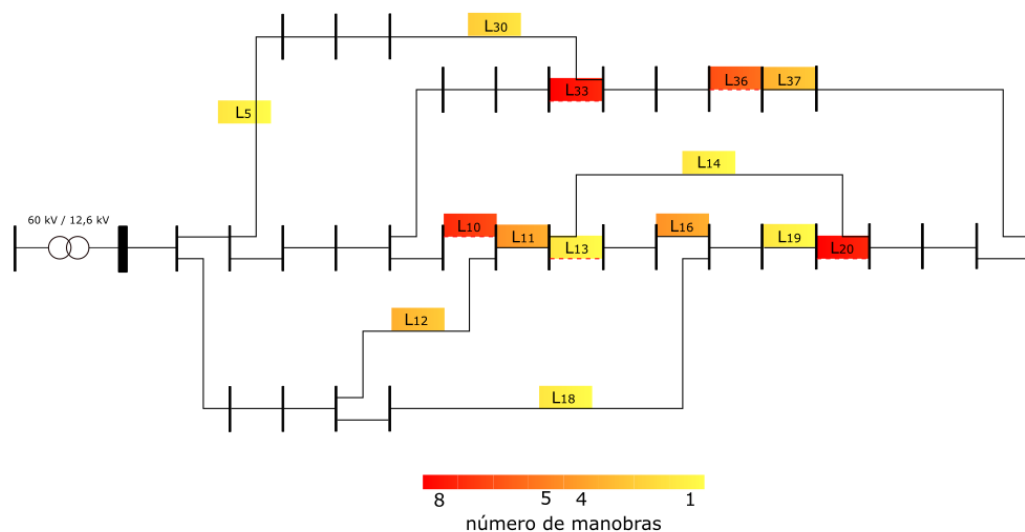


Figura 4.34: *Heatmap* que contém o número de manobras verificadas nos Cenário: Aumento de carga

Nas 10 configurações registadas para este cenário observou-se maior repetição nas linhas 10, 20, 33 e 36.

Para a localização de OCR's há que ter em consideração todos os cenários estudados, pois numa rede de distribuição de energia eléctrica é real que as cargas são variáveis ao longo do tempo.

Assim, é sugerida uma localização ideal de OCR's nos barramentos 8, 9, 12, 14, 25, 29, 32 e 33, obtendo um total de 8 manobras possíveis na rede, podendo assim atingir grande parte das configurações observadas ao longo dos diversos cenários de carga. A configuração final sugerida e a localização dos respectivos OCR's encontra-se na figura 4.35.

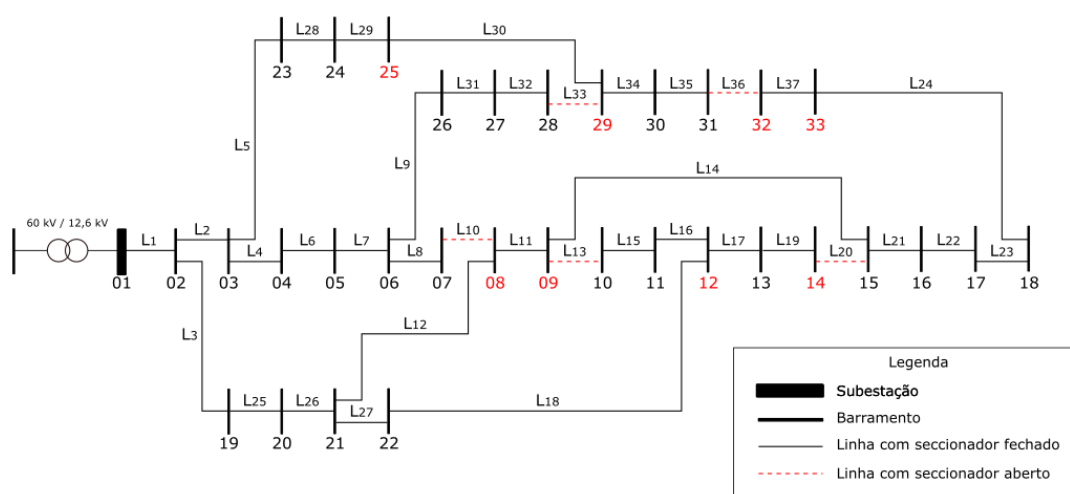


Figura 4.35: Configuração final e localização dos respectivos OCR's

4.4. PROPOSTA DE LOCALIZAÇÃO PREFERENCIAL DE OCR'S

Na Tabela 4.4 encontra-se consolidada a informação relativa à localização e estado preferencial dos OCR's e a percentagem de redução de perdas que é possível obter em cada cenário de carga.

Ramo onde colocar OCR	Estado preferencial	Redução de perdas (%)		
		Cenário: Base	Cenário: Redução de carga	Cenário: Aumento de carga
L10	0	37,9%	37,8%	37,8%
L13	0			
L16	1			
L20	0			
L24	1			
L30	1			
L33	0			
L36	0			

Tabela 4.4: Tabela com a localização e estado preferencial dos OCR's

Numa nota final, referir que, embora se tenha apenas utilizado o AG para analisar a localização preferencial dos OCR's, a configuração final da figura 4.35 obtida através do método de busca sequencial, encontra-se dentro das reconfigurações possíveis com a colocação dos OCR's indicados.

CONCLUSÕES

5.1 Objectivos alcançados e conclusões

O objectivo desta dissertação foi desenvolver uma metodologia que através de reconfiguração topológica possibilitasse uma minimização de perdas ajudando a identificar quais os locais adequados à colocação de OCR's em redes de distribuição de energia eléctrica.

Para o efeito, foi, inicialmente, desenvolvido um método de busca sequencial que, através da análise e reconfiguração de uma rede, procura de forma sequencial a configuração que apresenta menor valor de perdas por efeito de Joule. Embora este método tenha a vantagem de apresentar uma configuração que permite reduzir significativamente o valor de perdas de energia numa rede com tempos computacionais baixos para a rede de 33 barramentos em estudo, este método não tem a capacidade de fornecer várias topologias, uma vez que, o resultado final dependerá do ponto inicial de busca (i.e. a configuração inicial), ou seja, o método irá iniciar a busca a partir de uma das linhas que, inicialmente, se encontrem no estado aberto. Assim, a única maneira para se poder encontrar outras configurações seria alterando a configuração inicial da rede.

Posteriormente, devido ao carácter combinatório associado às manobras em redes, desenvolveu-se um algoritmo genético (AG) de forma a garantir a obtenção de resultados de elevada qualidade (valor de perdas reduzido), para os quais correspondam configurações de rede diferentes. Para o desenvolvimento do AG, adaptou-se o conceito de alguns operadores genéticos ao problema, para que a configuração da rede cumpra certos critérios.

A nível do cruzamento de indivíduos, recorreu-se a uma solução que permitisse o cruzamento dentro dos critérios impostos pelo problema. Para tal, o cruzamento entre indivíduos foi implementado comparando entre os dois membros se onde um tem uma linha aberta o outro tem a mesma linha fechada e vice-versa. Este método de cruzamento

desenvolvido podia por vezes não ser concretizável, dando acrescida importância à mutação como operador genético, uma vez que esta pode fornecer soluções que não seriam possíveis através do cruzamento.

De forma a aumentar a diversidade das soluções, reduzindo o tempo de convergência do algoritmo, foi proposto um tipo de mutação que permitiu atingir melhores resultados através da procura exaustiva na vizinhança de um indivíduo. A este operador genético chamou-se mutação inteligente. Ao desencadear-se a mutação inteligente, depois de escolhido aleatoriamente o indivíduo e a linha onde o seu estado será modificado de desligado para ligado, são exploradas todas as possibilidades de abertura de outra linha que verifique manter-se os critérios impostos pelo problema. A solução que dentro de todas as possibilidades tenha menor valor de perdas, será a vencedora desta mutação e irá juntar-se aos restantes indivíduos da população.

Toda a informação relativa ao desenvolvimento dos algoritmos encontra-se detalhadamente explicada e exemplificada no capítulo 3.

Finalmente, os algoritmos foram aplicados a uma rede de 33 barramentos e analisados três cenários de geração/carga. Conclui-se que através da colocação de 8 OCR's em locais preferenciais indicados é possível realizar manobras na rede reduzindo as perdas entre 35% a 38%, dependendo do cenário. Os resultados encontram-se em detalhe ao longo do capítulo 4.

Refira-se que para a identificação da localização preferencial dos OCR's necessitou-se recorrer à análise individual dos resultados de cada cenário de carga, escolhendo os locais onde ocorreram mais manobras, permitindo diversificação de soluções de reconfiguração em cada cenário. A escolha dos OCR's não pode ser alterada à medida que há variação de carga, assim, é imperativo obter uma solução que mantenha um bom nível de redução de perdas perante estas variações. Com a solução proposta, poder-se-á reconfigurar a rede de diferentes formas obtendo-se algumas das melhores configurações encontradas em cada cenário.

Deste modo, conclui-se que o AG desenvolvido é adequado à optimização da localização dos OCR's capazes de reconfigurar uma rede de distribuição de energia eléctrica, reduzindo, consequentemente, as perdas activas.

5.2 Trabalho futuro

O trabalho desenvolvido ao longo desta dissertação abre várias hipóteses de trabalho futuro. Destacam-se os seguintes:

- Aplicação do algoritmo genético a casos de maior dimensão;
- Estudo paramétrico do algoritmo genético de modo a melhorar o seu desempenho, qualidade e diversidade das soluções obtidas;

- Inclusão de custos no algoritmo de busca de forma a obter soluções tecnicamente viáveis e economicamente sustentáveis (relação custo/benefício associado à instalação de OCR's) para uma qualquer rede de distribuição.

BIBLIOGRAFIA

- [1] 2019. URL: <http://www.erse.pt/pt/electricidade/actividadesdosector/distribuicao/Paginas/RND-Perdas.aspx>.
- [2] 2019. URL: <http://www.erse.pt/pt/electricidade/actividadesdosector/transporte/Paginas/default.aspx>.
- [3] Alexandre Dias, Under Supervision of Prof. João Santana and Prof. Pedro Carvalho. *Economic Regulation of Losses in the Electricity Distribution Network*. URL: <https://fenix.tecnico.ulisboa.pt/downloadFile/395145025355/resumo.pdf>.
- [4] A. Álzate, O. D. Montoya, R. A. Hincapié e M. Granada. “Optimal location of reclosers in distribution systems considering reliability in communication channels”. Em: *2015 IEEE 6th Latin American Symposium on Circuits and Systems, LASCAS 2015 - Conference Proceedings* (2015), pp. 2–5. DOI: [10.1109/LASCAS.2015.7250487](https://doi.org/10.1109/LASCAS.2015.7250487).
- [5] S. Bisanovic, M. Samardzic e D. Aganovic. “Economic valuation of power and energy losses in distribution networks”. Em: *International Journal of Electrical and Computer Engineering* 6.2 (2016), pp. 439–446. ISSN: 20888708. DOI: [10.11591/ijece.v6i2.9182](https://doi.org/10.11591/ijece.v6i2.9182).
- [6] P. M. S. Carvalho, L. A.F. M. Ferreira e A. J. C. da Silva. “A decomposition approach to optimal remote controlled switch allocation in distribution systems”. Em: *IEEE Transactions on Power Delivery* 20.2 (abr. de 2005), pp. 1031–1036. ISSN: 0885-8977. DOI: [10.1109/TPWRD.2004.838470](https://doi.org/10.1109/TPWRD.2004.838470).
- [7] Diogo Filipe Rodrigues Antunes da Costa. “Sistema de Monitorização de Qualidade de Sinal GSM/GPRS para a eficiência do Telecomando da Rede de Média Tensão”. Tese de mestrado. Dez. de 2015.
- [8] *Dispositivos Automáticos para Isolamento de Defeitos e Reposição de Serviço*. URL: <https://paginas.fe.up.pt/~ee99017/IAT.PDF>.
- [9] E. Distribuição. “Materiais para redes – Aparelhagem AT e MT. Órgão de corte de rede tipo 1 - OCR 1”. Em: (fev. de 2007).
- [10] E. Distribuição. “Materiais para redes – Aparelhagem AT e MT. Órgão de corte de rede tipo 2 - OCR 2”. Em: (mai. de 2007).
- [11] E. Distribuição. “Materiais para redes – Aparelhagem AT e MT. Órgão de corte de rede tipo 3 - OCR 3”. Em: (set. de 2014).

- [12] J. W. Fourie e J. E. Calmeyer. “A statistical method to minimize electrical energy losses in a local electricity distribution network”. Em: 2 (set. de 2004), pp. 667–673. DOI: [10.1109/AFRICON.2004.1406768](https://doi.org/10.1109/AFRICON.2004.1406768).
- [13] E. Jordanger, K. Sand e R. Kristensen. “Method for calculation of cost of electrical power system losses”. Em: 5 (2001). ISSN: 0537-9989. DOI: [10.1049/cp:20010876](https://doi.org/10.1049/cp:20010876).
- [14] João Francisco Lopes Ferreira. “Os Órgãos de Corte de Rede de 3º Geração nas Redes de Distribuição de Média Tensão”. Tese de mestrado. Jan. de 2013.
- [15] A. H. Kadhim. “Energy Loss Reduction in Distribution System”. Em: *International Journal of Science and Research (IJSR)* 4.2 (2015), pp. 1248–1251. ISSN: 2319-7064. URL: <https://www.ijsr.net/archive/v4i2/SUB151225.pdf>.
- [16] J. P. S. Paiva. *Redes de Energia Eléctrica: Uma Análise Sistémica*. 4ª ed. Mai. de 2015. ISBN: 978-989-8481-41-2.
- [17] Pedro Miguel Lopes Torres. “Metodologia para melhoria de perdas e fiabilidade em redes de distribuição”. Tese de mestrado. Instituto Superior de Engenharia de Lisboa, 2010.
- [18] N. Soni e T. Kumar. “Study of Various Mutation Operators in Genetic Algorithms”. Em: 5.3 (2014), pp. 4519–4521.
- [19] R Srinivasa Rao e S. Narasimham. “A new heuristic approach for optimal network reconfiguration in distribution systems”. Em: *World Academy of Science, Engineering and Technology* 3.9 (2009), pp. 2240–2246. ISSN: 2010-3778.
- [20] D. I. H. Sun, S. Abe, R. R. Shoults, M. S. Chen, P. Eichenberger e D. Farris. “Calculation of Energy Losses in a Distribution System”. Em: *IEEE Transactions on Power Apparatus and Systems* PAS-99.4 (jul. de 1980), pp. 1347–1356. ISSN: 0018-9510. DOI: [10.1109/TPAS.1980.319557](https://doi.org/10.1109/TPAS.1980.319557).
- [21] T. Wagner. “Impact of remote controlled switches on distribution grid recovering”. Tese de mestrado. 2010.
- [22] L. Zhang, Y. Xu e Z. Wang. “Research on optimization of recloser placement of DG-enhanced distribution networks”. Em: *3rd International Conference on Deregulation and Restructuring and Power Technologies, DRPT 2008 April* (2008), pp. 2592–2597.

A N E X O



CÓDIGO DESENVOLVIDO

I.1 Funções essenciais do método de busca sequencial

1. Funções de inicialização dos mapas da rede

```
1  #-----
2  # Mapa de todas as linhas existentes numa rede
3  #-----
4  def initialize_map():
5      # Regista o número total de barramentos
6      ierr, Nbuses = psspy.abuscount(flag=2)
7      # Cria mapa com Nbarramentos+1 por Nbarramentos+1
8      the_map = np.zeros((Nbuses+1,Nbuses+1))
9
10     for i in range(1, Nbuses+1):
11         #Inicializa a iteração para cada BUS
12         ierr = psspy.inibrn(i, 2)
13         while True:
14             #Retorna todas as opções
15             ierr, jbus, ickt = psspy.nxtbrn(i)
16             if ierr > 0:
17                 break
18             else:
19                 #Verifica se é transformador, se não for adiciona linha, caso contrário
20                 #ignora
21                 ierr, model = psspy.cctmdlbnam_brnd(i, jbus, ickt)
22                 if(ierr != 5):
23                     ierr, ival = psspy.brnint(i, jbus, ickt, 'STATUS')
24                     if ival == 1:
25                         #Coloca 1 onde existem linhas com SF & -1 onde existem linhas
26                         #com SA
27                         the_map[i][jbus] = 1
28                         the_map[jbus][i] = the_map[i][jbus]
29                     else:
30                         the_map[i][jbus] = -1
31                         the_map[jbus][i] = the_map[i][jbus]
32                 else:
33                     continue
34         continue
35     return the_map
36
37 #-----
38 # Matriz que contém os valores de queda de tensão para todas as linhas abertas
39 #-----
40 def ddp(the_map):
41     # Regista o número total de barramentos
42     ierr, Nbuses = psspy.abuscount(flag=2)
43     # Cria mapa ddp com Nbarramentos+1 por Nbarramentos+1
44     ddp_map = np.zeros((Nbuses+1,Nbuses+1))
45
46     for i in range(1, len(the_map)):
47         for j in range(1, len(the_map)):
48             # Encontra linha com SA e regista a queda de tensão na mesma
49             if (the_map[i][j] == -1):
50                 ierr, rval1 = psspy.busdat(i,'PU')
51                 ierr, rval2 = psspy.busdat(j,'PU')
52                 ddp_map[i][j] = abs(rval1 - rval2)
53                 ddp_map[j][i] = ddp_map[i][j]
54     return ddp_map
```

2. Funções essenciais do algoritmo de busca sequencial

```
1  #-----
2  # Localiza qual a linha com maior DDP e retorna o IBUS, JBUS e o valor da mesma
3  #-----
4  def max_ddp(ddp_map):
5      value_old = 0
6      inew = 0
7      jnew = 0
8      value_new = 0
9      for i in range (1, len(ddp_map)):
10         for j in range (1, len(ddp_map)):
11             if ddp_map[i][j] > value_old :
12                 value_new = ddp_map[i][j]
13                 value_old = value_new
14                 inew = i
15                 jnew = j
16         return inew, jnew, value_new
17
18  #-----
19  # Retorna o número do barramento que tem menor valor de tensão para a linha
20  # anteriormente seleccionada;
21  # Recebe barramentos que se encontram na extremidade da linha
22  #-----
23  def min_bus(ibus, jbus):
24      ierr, rval1 = psspy.busdat(ibus, 'PU')
25      ierr, rval2 = psspy.busdat(jbus, 'PU')
26
27      if (rval1 - rval2) < 0 :
28          min_bus_value = ibus
29      else:
30          min_bus_value = jbus
31
32      return min_bus_value
33
34  #-----
35  # Verifica se é possível desligar linha adjacente sem criar ilhas na rede;
36  # Recebe barramento a desligar e qual o barramento que foi desligado anteriormente
37  #-----
38  def valid_branch(ibus, rede, bus_anterior):
39      buses = 0
40      ioff = 0
41      joff = 0
42      for j in range (1, len(rede)):
43          if (j != bus_anterior):
44              if (rede[ibus][j] == 1):
45                  line_status(ibus, j, 0, rede)
46                  print '\n\t\t\tBranch from ' + str(ibus) + ' to ' + str(j) + ' is
47                  now out-of-service\n'
48                  # Processo de verificação de ilhas .tree
49                  ierr, buses = psspy.tree(1)
50                  print buses
51                  if (buses <> 0):
52                      # Existem ilhas
53                      ierr, buses = psspy.tree(2, -1)
54                      line_status(ibus, j, 1, rede)
55                      print '\n\t\t\tBranch from ' + str(ibus) + ' to ' + str(j) + '
56                      is back on-service\n'
57                  else:
58                      # Não existem ilhas
59                      line_status(ibus, j, 1, rede)
60                      ioff = ibus
61                      joff = j
62              else:
63                  continue
64          else:
65              continue
66      buses = 0
67      # Se não houver problemas desliga a linha
68      if(ioff <> 0 and joff <> 0):
69          line_status(ioff, joff, 0, rede)
70          return joff
71      # Se houver problemas não desliga a linha e retorna 0 (impossível)
72      else:
73          return 0
```

```

70         return 0
71
72 #-----
73 # Tr nsito de energia usando o m todo de Newton-Raphson
74 #-----
75 def solve_case():
76     # Solve Power Flow with Newton-Raphson Method
77     ierr = psspy.fns1([[0, 0, 0, 1, 1, 1, 0, 0]])
78
79     if ierr == 3:
80         print("\n\n\t\t\t BUSES in ISLAND(s)!\n\n")
81         return 3
82     if ierr < 0:
83         print("\n\n\t\t\t Solving Error...\n\n")
84         print(ierr)
85     else:
86         print("\n\n\t\t\t Successful Simulation!\n\n")
87
88     err = psspy.solved() # Check if the last solution reached tolerance
89     if err < 0:
90         print "\n\nError! - Tolerance not reached!\n\n"
91         print err
92     else:
93         print "\n\nSolve OK - Tolerance reached.\n\n"
94
95 #-----
96 # Calcula total de perdas activas na rede
97 #-----
98 def total_losses():
99     # tr nsito de energia
100     solve_case()
101
102     # ONLY BRANCHES LOSSES
103     # perdas activas
104     ierr, BRN_Ploss = psspy.abrreal(-1, entry=1, string="PLOSS")
105     # perdas reactivas
106     ierr, BRN_Qloss = psspy.abrreal(-1, entry=1, string="QLOSS")
107     # total de barramentos
108     ierr, Nbranches = psspy.abrncount(-1, entry=1)
109
110     tot_ploss = 0.0
111     tot_qloss = 0.0
112
113     for k in range (0, Nbranches):
114         tot_ploss += float(BRN_Ploss[0][k])
115         tot_qloss += float(BRN_Qloss[0][k])
116
117     # retorna valor total de perdas activas na rede
118     return tot_ploss
119
120 #-----
121 # Altera o estado da linha (0 - desligada, 1 - ligada)
122 #-----
123 def line_status(ibus, jbus, estado, rede):
124     ierr = psspy.branch_chng_3(ibus, jbus, '1', intgar=estado)
125     if estado == 1:
126         rede[ibus,jbus] = 1
127         rede[jbus,ibus] = 1
128     if estado == 0:
129         rede[ibus,jbus] = 0
130         rede[jbus,ibus] = 0
131     return ierr

```


I.2 Funções essenciais do Algoritmo Genético

1. Operador genético - Crossover

```
1  #-----
2  # Operador genético - CROSSOVER
3  # Recebe as duas populações (1 e 2) que se irão cruzar
4  #-----
5  def crossover(pop1, pop2):
6      # variáveis
7      new_fitness = []
8      last_i = 0
9      last_j = 0
10     finished = 0
11     i_fechar = 0
12     j_fechar = 0
13     crossover = 1
14     max_range = len(pop1)
15     new_individual = pop1.copy()
16     fechar = 1
17     abrir = 0
18     n_abrir = 0
19     n_fechar = 0
20     cont_abrir = 0
21     cont_fechar = 0
22
23     while finished != 1:
24         print "CROSSOVER - while\n"
25         crossover = 1
26         cont_fechar = 0
27         cont_abrir = 0
28         print "i fechar: " + str(i_fechar) + " j fechar: " + str(j_fechar) + " i abrir: "
29           + str(last_i) + " j abrir: " + str(last_j)
30         while crossover == 1:
31             for i in range(1, max_range):
32                 if crossover == 1:
33                     for j in range(i, max_range):
34                         print "i: " + str(i) + " j: " + str(j) + " n fechar: " +
35                           str(n_fechar) + " n abrir: " + str(n_abrir) + " cont fechar: "
36                           + str(cont_fechar) + " cont abrir: " + str(cont_abrir)
37                         # verifica condição para fechar linha
38                         if(fechar == 1 and pop1[i][j] == 1 and pop2[i][j] == 2 and
39                           n_fechar == cont_fechar):
40                             print "CROSSOVER - fechar\n"
41                             new_individual[i][j] = 2
42                             new_individual[j][i] = 2
43                             i_fechar = i
44                             j_fechar = j
45                             crossover = 0
46                             abrir = 1
47                             fechar = 0
48                             n_fechar += 1
49                             cont_fechar += 1
50                             break
51                         if(fechar == 1 and pop1[i][j] == 1 and pop2[i][j] == 2 and
52                           n_fechar != cont_fechar):
53                             cont_fechar += 1
54                             continue
55
56                         # verifica condição para abrir linha
57                         if(abrir == 1 and pop1[i][j] == 2 and pop2[i][j] == 1 and
58                           n_abrir == cont_abrir):
59                             print "CROSSOVER - abrir\n"
60                             last_i = i
61                             last_j = j
62                             new_individual[i][j] = 1
63                             new_individual[j][i] = 1
64                             n_abrir += 1
65                             cont_abrir += 1
66                             island = island_check(new_individual)
67                             if(island == 0):
```

```

62         finished = 1
63         crossover = 0
64         new_fitness = total_losses(copy.copy(new_individual))
65         return new_individual, new_fitness
66     if(island == 1):
67         new_individual[i][j] = 2
68         new_individual[j][i] = 2
69         crossover = 0
70         break
71
72     if(abrir == 1 and pop1[i][j] == 2 and pop2[i][j] == 1 and
73        n_abrir != cont_abrir):
74         cont_abrir += 1
75         continue
76
77     # Caso não seja possível abrir nenhuma linha
78     if(abrir == 1 and i == max_range-1 and j == max_range-1):
79         print "NAO FOI POSSIVEL ENCONTRAR POS DE ABRIR, TENTAR
80         FECHAR NOVAMENTE\n"
81         new_individual[i_fechar][j_fechar] = 1
82         new_individual[j_fechar][i_fechar] = 1
83         fechar = 1
84         abrir = 0
85         crossover = 0
86         break
87
88     # Caso não seja possível realizar cruzamento
89     if(fechar == 1 and i == max_range-1 and j == max_range-1):
90         print "NAO FOI POSSIVEL ENCONTRAR POS DE FECHAR, NAO E
91         POSSIVEL CRUZAR\n"
92         #island = island_check(new_individual)
93         new_fitness = total_losses(copy.copy(new_individual))
94         return new_individual, new_fitness
95     else:
96         continue
97     break
98
99     return new_individual, new_fitness

```

2. Operador genético: Mutação Inteligente

```
1  #-----
2  # Operador genético - MUTAÇÃO INTELIGENTE
3  # Recebe toda a população e vê qual será aleatoriamente mutada,
4  # para tal é gerado um valor aleatoriamente de 0 a 1, que tem de ser menor do que a
5  # chance de haver mutação (int_mut_chance)
6  #-----
7  def intelligent_mutate(mut_pop, mut_fitness, int_mut_chance, nClose, nOpen):
8      # variaveis
9      global cont_ger
10     global cont_intmut
11     global hist_int_mut
12     global writer
13     cont_valida = 0
14
15     intmut_pop = copy.deepcopy(mut_pop)
16     intmut_fitness = copy.deepcopy(mut_fitness)
17
18     print "Inicio Mutação inteligente\n"
19
20     for p in range(0, len(intmut_pop)):
21         temp_pop = []
22         temp_fitness = []
23         mapa = []
24         print "POPULAÇÃO NUMERO {P}: " + str(p) + "\n"
25
26         fechar = 1
27         abrir = 0
28         loop = 1
29         go = 1
30         contagem_fechar = 1
31         i_fechado = 0
32         j_fechado = 0
33
34         mapa = copy.deepcopy(intmut_pop[p])
35         print "fitness inicial: " + str(total_losses(intmut_pop[p]))
36
37         if random.random() < int_mut_chance:
38             print "entrou na chance de mut inteligente"
39             cont_intmut += 1
40
41             print "Fazer mutação da população numero: " + str(p)
42
43             # selecciona aleatoriamente uma linha para fechar
44             i_fechar = np.random.randint(1, nOpen+1)
45             print "fechar linha nº: " + str(i_fechar)
46
47             for i in range(1, len(mapa)):
48                 for j in range(i, len(mapa)):
49                     if (mapa[i][j] == 1 and fechar == 1):
50                         print "Linha aberta.Contagem: " + str(contagem_fechar) + ".
51                         Fechar esta linha"
52                         if i_fechar == contagem_fechar:
53                             print "encontrou linha para fechar\n"
54                             mapa[i][j] = 2
55                             mapa[j][i] = 2
56
57                             i_fechado = i
58                             j_fechado = j
59
60                             print "i fechado = " + str(i_fechado) + " j fechado = "
61                             + str(j_fechado)
62
63                             loop = 0
64                             fechar = 0
65                             abrir = 1
66                             break
67                         else:
68                             contagem_fechar += 1
69                             continue
70             if loop == 0:
71                 break
```

```

70     for i in range(1, len(mapa)):
71         for j in range(i, len(mapa)):
72             if (mapa[i][j] == 2 and (i != i_fechado or j != j_fechado) and
73                 fechar == 0 and abrir == 1):
74                 print "Linha fechada. Abrir esta linha"
75                 print "encontrou linha para abrir, i = " + str(i) + ", j = "
76                     + str(j) + "\n"
77                 mapa[i][j] = 1
78                 mapa[j][i] = 1
79
80                 i_aberto = i
81                 j_aberto = j
82
83                 island = island_check(mapa)
84
85                 if (island == 0):
86                     cont_valida += 1
87                     print "Solução valida, i aberto : " + str(i_aberto) + "
88                         j aberto: " + str(j_aberto)
89
90                     # Guardar solução temporária
91                     temp_pop.append(copy.deepcopy(mapa))
92                     temp_fitness.append(copy.deepcopy(total_losses(mapa)))
93
94                     # Voltar a fechar linha
95                     print "voltar a fechar a linha que foi aberta, i = " +
96                         str(i) + ", j = " + str(j) + "\n"
97                     mapa[i_aberto][j_aberto] = 2
98                     mapa[j_aberto][i_aberto] = 2
99
100                     continue
101
102                     if (island == 1):
103                         print "Existe ilha\n"
104                         print "Solução inválida. Voltar a fechar linha\n"
105                         mapa[i_aberto][j_aberto] = 2
106                         mapa[j_aberto][i_aberto] = 2
107                         continue
108                     else:
109                         continue
110
111     if not temp_fitness:
112         print "Não foi possível achar soluções possíveis\n"
113         # Registo de dados mutação inteligente
114         hist_int_mut[0, cont_intmut] = cont_ger
115         hist_int_mut[1, cont_intmut] = cont_intmut
116         hist_int_mut[2, cont_intmut] = p
117         hist_int_mut[3, cont_intmut] = None
118         hist_int_mut[4, cont_intmut] = intmut_fitness[p]
119         hist_int_mut[5, cont_intmut] = None
120     else:
121         # selecciona o menor valor de fitness
122         min_fitness = np.amin(temp_fitness)
123
124         # percorre todas as soluções
125         for i in range(0, len(temp_fitness)):
126             print "temp fitness: " + str(i) + "valor: " + str(temp_fitness[i])
127             if temp_fitness[i] == min_fitness:
128                 #print "max value, cross and fitness pop\n"
129                 min_fit_num = i
130
131                 # ao encontrar a melhor adiciona-a à população
132                 intmut_fitness.append(copy.deepcopy(min_fitness))
133                 intmut_pop.append(copy.deepcopy(temp_pop[i]))
134
135                 print "fitness added: " + str(min_fitness) + " posição: " +
136                     str(i)
137                 break
138             else:
139                 continue
140
141     # Registo de dados mutação inteligente

```

```

137         hist_int_mut[0, cont_intmut] = cont_ger
138         hist_int_mut[1, cont_intmut] = cont_intmut
139         hist_int_mut[2, cont_intmut] = p
140         hist_int_mut[3, cont_intmut] = len(temp_fitness)
141         hist_int_mut[4, cont_intmut] = intmut_fitness[p]
142         hist_int_mut[5, cont_intmut] = min_fitness
143
144         print "contagem valida: " + str(cont_valida)
145
146     else:
147         # Não entrou na chance de haver mutação inteligente
148         print "else de nao entrou em mutacao"
149         continue
150
151     print "save excel"
152     writer_int.save()
153
154     # retorna a população já com o novo individuo gerado através da mutação
155     # inteligente
156     return intmut_pop, intmut_fitness

```

3. Registro das melhores configurações (*vector_best*)

```
1  #-----
2  # Função que verifica todos os indivíduos da população e adiciona os melhores
3  # até preencher o vector com o parametro (tamanho) que foi escolhido anteriormente
4  #-----
5  def choose_best(best_size, best_pop, best_fitness, new_pop, new_fitness):
6      tempo_inicio = time.time()
7
8      fitness_copy = copy.copy(new_fitness)
9      pop_copy = copy.copy(new_pop)
10
11     print "comprimento fitness_copy: " + str(len(fitness_copy))
12
13     # caso ainda não existam soluções no vector
14     if len(best_fitness) == 0 :
15         print "best fitness vazio\n"
16         while ((len(best_fitness) < best_size) and (len(fitness_copy) > 0)):
17             print fitness_copy
18             min_fitness = np.amin(fitness_copy)
19
20             # percorre todos os membros da população
21             for i in range(0, len(fitness_copy)):
22                 if fitness_copy[i] == min_fitness:
23
24                     best_fitness.append(min_fitness)
25                     best_pop.append(copy.copy(pop_copy[i]))
26                     save_case(pop_copy[i])
27                     print "fitness added: " + str(min_fitness)
28                     fitness_copy.pop(i)
29                     pop_copy.pop(i)
30                     break
31             else:
32                 continue
33     # caso já existam soluções no vector
34     else:
35         while (len(fitness_copy) > 0):
36             print "len fitness copy: " + str(len(fitness_copy))
37             print "len best fitness: " + str(len(best_fitness))
38
39             compare_vector = np.intersect1d(fitness_copy, best_fitness)
40             print "compare vector: " + str(compare_vector)
41
42             min_fitness = amin(fitness_copy)
43             max_best = np.amax(best_fitness)
44             print "min fit: " + str(min_fitness) + "\nmax fitness: " + str(max_best)
45
46             compare = np.intersect1d(min_fitness, compare_vector)
47             print "compare: " + str(compare)
48
49             # a solução já existe no vector
50             if compare != 0 :
51                 for i in range(0, len(fitness_copy)):
52                     if fitness_copy[i] == min_fitness:
53                         fitness_copy.pop(i)
54                         pop_copy.pop(i)
55                         break
56             else:
57                 continue
58
59     # a solução ainda não existe no vector e o vector não se encontra cheio
60     elif len(best_fitness) < best_size and len(compare) == 0:
61         print "best fitness not full\n"
62         for i in range(0, len(fitness_copy)):
63             if fitness_copy[i] == min_fitness:
64                 best_fitness.append(min_fitness)
65                 best_pop.append(copy.copy(pop_copy[i]))
66                 print "fitness added: " + str(min_fitness)
67                 fitness_copy.pop(i)
68                 pop_copy.pop(i)
69                 break
70             else:
71                 continue
72     # a solução ainda não existe no vector e o vector mas este encontra-se
```

```

cheio, sendo necessário substituir caso a solução seja melhor que a pior
que se encontra no vector
73 elif len(best_fitness) == best_size and len(compare) == 0 and
min_fitness < max_best:
74     print "best fitness full, substituir valor"
75     for i in range(0, len(best_fitness)):
76         if best_fitness[i] == max_best:
77             best_fitness.pop(i)
78             best_pop.pop(i)
79             print "best fitness popped: " + str(max_best)
80             break
81         else:
82             continue
83     for i in range(0, len(fitness_copy)):
84         if fitness_copy[i] == min_fitness:
85             best_fitness.append(min_fitness)
86             best_pop.append(copy.copy(pop_copy[i]))
87             print "fitness added: " + str(min_fitness)
88             fitness_copy.pop(i)
89             pop_copy.pop(i)
90             break
91         else:
92             continue
93     else:
94         for i in range(0, len(fitness_copy)):
95             if fitness_copy[i] == min_fitness :
96                 fitness_copy.pop(i)
97                 pop_copy.pop(i)
98                 print "fitness popped: " + str(min_fitness)
99                 break
100
101 tempo_best = time.time() - tempo_inicio
102 return best_pop, best_fitness, tempo_best
103

```


A N E X O



REDE TESTE DE 8 BARRAMENTOS

II.1 Características técnicas aplicadas à rede teste

Nº Linha	Barramentos		Características das linhas			Switch	Estado do switch
	de	para	R (Ω)	X (Ω)	L (km)		
1	1	2	0,2030	0,1034	0,6	0	0
2	2	3	0,2030	0,1034	0,6	1	1
3	2	5	0,2030	0,1034	0,6	1	1
4	3	4	0,2030	0,1034	0,6	1	1
5	3	6	0,2030	0,1034	0,6	1	0
6	4	8	0,2030	0,1034	0,6	1	0
7	5	6	0,2030	0,1034	0,6	1	1
8	6	7	0,2030	0,1034	0,6	1	1
9	7	8	0,2030	0,1034	0,6	1	1

Barramentos	Carga Ativa (MW)	Carga Reativa (MVAR)	tg φ (°)
2	1,20	0,60	0,50
3	0,50	0,10	0,20
4	0,50	0,10	0,20
5	0,50	0,10	0,20
6	1,00	0,50	0,50
7	2,00	1,00	0,50
8	0,50	0,10	0,20

II.2 Resultado método de busca sequencial - Rede teste

Iteração		Inicial	1	2	3	4
Linha	Ligada	--	6	9	5	7
	Desligada	--	9	8	7	3
perdas (kW)		110,6975	100,5405	108,1369	108,8972	119,8844
Tensão nodal (p.u) nos barramentos	1	1	1		1	
	2	0,990534	0,990534		0,990534	
	3	0,989139	0,988441		0,983674	
	4	0,988441	0,987045		0,982278	
	5	0,984371	0,985069		0,989837	
	6	0,978906	0,980302		0,978906	
	7	0,97503	0,977123		0,975728	
	8	0,974332	0,986348		0,98158	

A N E X O



REDE DE 33 BARRAMENTOS

III.1 Características das linhas da rede de 33 barramentos

Nº Linha	Barramentos		Características das linhas			Switch	Estado do switch
	de	para	R (Ω)	X (Ω)	L (km)		
1	1	2	0,09220	0,04770	0,3	0	0
2	2	3	0,49300	0,25110	1,5	1	1
3	2	19	0,16400	0,15650	0,5	1	1
4	3	4	0,36600	0,18400	1,1	1	1
5	3	23	0,45120	0,30830	1,3	1	1
6	4	5	0,38110	0,19410	1,1	1	1
7	5	6	0,81900	0,07000	2,4	1	1
8	6	7	0,18720	0,61880	0,6	1	1
9	6	26	0,20300	0,10340	0,6	1	1
10	7	8	1,71140	1,23510	5,1	1	1
11	8	9	1,03000	0,74000	3,0	1	1
12	8	21	0,00000	2,00000	5,9	1	0
13	9	10	1,04000	0,74000	3,1	1	1
14	9	15	0,00000	2,00000	5,9	1	0
15	10	11	0,19660	0,06500	0,6	1	1
16	11	12	0,37440	0,12380	1,1	1	1
17	12	13	1,46800	1,15500	4,3	1	1
18	12	22	0,00000	2,00000	5,9	1	0
19	13	14	0,54160	0,71290	1,6	1	1
20	14	15	0,59100	0,52600	1,7	1	1
21	15	16	0,74630	0,54500	2,2	1	1
22	16	17	1,28900	1,72100	3,8	1	1
23	17	18	0,73200	0,57400	2,2	1	1
24	18	33	0,00000	2,00000	5,9	1	0
25	19	20	1,50420	1,35540	4,4	1	1
26	20	21	0,40950	0,47840	1,2	1	1
27	21	22	0,70890	0,93730	2,1	1	1
28	23	24	0,89800	0,70910	2,7	1	1
29	24	25	0,89600	0,70110	2,6	1	1
30	25	29	0,00000	2,00000	5,9	1	0
31	26	27	0,28420	0,14470	0,8	1	1
32	27	28	1,05900	0,93370	3,1	1	1
33	28	29	0,80420	0,70060	2,4	1	1
34	29	30	0,50750	0,25850	1,5	1	1
35	30	31	0,97440	0,96300	2,9	1	1
36	31	32	0,31050	0,36190	0,9	1	1
37	32	33	0,34100	0,53020	1,0	1	1

Assumindo que as linhas são constituídas por material da marca CABELTE do tipo ACSR, tabela ASTM B-32 standard, code word: PIGEON e que apresentam a 25°C(a.c) uma resistencia de 0,3386 Ω /km (ver anexo 24)

III.2 Cenários de Carga aplicados à rede de 33 barramentos

Barramentos	Cenário Normal			Cenário +20% carga			Cenário -20% carga			tg φ (°)
	Carga Ativa (MW)	Carga Reativa (MVar)		Carga Ativa (MW)	Carga Reativa (MVar)		Carga Ativa (MW)	Carga Reativa (MVar)		
2	0,10	0,06		0,120	0,072		0,080	0,048		0,60
3	0,09	0,04		0,108	0,048		0,072	0,032		0,44
4	0,12	0,08		0,144	0,096		0,096	0,064		0,67
5	0,06	0,03		0,072	0,036		0,048	0,024		0,50
6	0,06	0,02		0,072	0,024		0,048	0,016		0,33
7	0,20	0,10		0,240	0,120		0,160	0,080		0,50
8	0,20	0,10		0,240	0,120		0,160	0,080		0,50
9	0,06	0,02		0,072	0,024		0,048	0,016		0,33
10	0,06	0,02		0,072	0,024		0,048	0,016		0,33
11	0,05	0,03		0,054	0,036		0,036	0,024		0,67
12	0,06	0,04		0,072	0,042		0,048	0,028		0,58
13	0,06	0,04		0,072	0,042		0,048	0,028		0,58
14	0,12	0,08		0,144	0,096		0,096	0,064		0,67
15	0,06	0,01		0,072	0,012		0,048	0,008		0,17
16	0,06	0,02		0,072	0,024		0,048	0,016		0,33
17	0,06	0,02		0,072	0,024		0,048	0,016		0,33
18	0,09	0,04		0,108	0,048		0,072	0,032		0,44
19	0,09	0,04		0,108	0,048		0,072	0,032		0,44
20	0,09	0,04		0,108	0,048		0,072	0,032		0,44
21	0,09	0,04		0,108	0,048		0,072	0,032		0,44
22	0,09	0,04		0,108	0,048		0,072	0,032		0,44
23	0,09	0,05		0,108	0,060		0,072	0,040		0,56
24	0,42	0,20		0,504	0,240		0,336	0,160		0,48
25	0,42	0,20		0,504	0,240		0,336	0,160		0,48
26	0,06	0,03		0,072	0,030		0,048	0,020		0,42
27	0,06	0,03		0,072	0,030		0,048	0,020		0,42
28	0,06	0,02		0,072	0,024		0,048	0,016		0,33
29	0,02	0,07		0,024	0,084		0,016	0,056		3,50
30	0,20	0,60		0,240	0,720		0,160	0,480		3,00
31	0,15	0,07		0,180	0,084		0,120	0,056		0,47
32	0,21	0,10		0,252	0,120		0,168	0,080		0,48
33	0,06	0,04		0,072	0,048		0,048	0,032		0,67

ANEXO
IV

CONDUTORES LINHAS AÉREAS CABELTE

CENELEC EN50182:2001 standard

Characteristics of aluminium conductors steel reinforced – Type AL1 / ST6C – France

Code number	Old code	Cross-section			Composition				Nominal diameter		Linear mass (kg/km)	Rated strength (kN)	Electrical resistance d.c. 20°C (Ω/km)	Stranding direction
		Alum. (mm ²)	Steel (mm ²)	Total (mm ²)	Aluminium		Steel		Core (mm)	Cond. (mm)				
					N.º of wires	Diam. (mm)	N.º of wires	Diam. (mm)						
94-AL1/22-ST6C	CROCUS 116,2	94,2	22,0	116,2	30	2,00	7	2,00	6,00	14,0	432,5	49,32	0,3067	S
119-AL1/28-ST6C	CROCUS 147,1	119,3	27,8	147,1	30	2,25	7	2,25	6,75	15,8	547,4	61,83	0,2423	S
147-AL1/34-ST6C	CROCUS 181,6	147,3	34,4	181,6	30	2,50	7	2,50	7,50	17,5	675,8	74,22	0,1963	S
185-AL1/43-ST6C	CROCUS 228	184,7	43,1	227,8	30	2,80	7	2,80	8,40	19,6	847,7	92,18	0,1565	S
234-AL1/55-ST6C	CROCUS 288	233,8	54,6	288,3	30	3,15	7	3,15	9,45	22,1	1 072,8	113,86	0,1236	S
222-AL1/76-ST6C	CROCUS 297	221,7	75,5	297,2	36	2,80	19	2,25	11,3	22,5	1 206,8	147,22	0,1307	Z
326-AL1/86-ST6C	CROCUS 412	325,7	86,0	411,7	32	3,60	19	2,40	12,0	26,4	1 576,1	173,31	0,0889	Z
508-AL1/105-ST6C	CROCUS 612	507,8	104,8	612,6	66	3,13	19	2,65	13,3	32,0	2 226,5	231,55	0,0570	S
717-AL1/148-ST6C	CROCUS 865	717,3	148,1	865,4	66	3,72	19	3,15	15,8	38,1	3 145,4	319,11	0,0403	S
957-AL1/228-ST6C	CROCUS 1 185	956,7	227,8	1 184,5	54	2,80	37	2,80	19,6	44,7	4 433,6	480,75	0,0302	S

Note – Values of final modulus of elasticity and coefficient of linear expansion for the conductor sizes listed in the Table are used in France. Values for other conductor constructions may be calculated using the method given in IEC 61597.

ASTM B-232 standard

Characteristics of aluminium conductors steel reinforced

Code word	Size (AWG or kcmil)	Composition		Cross-section		Outer diameter		Cable weight			Rated strength (N)	Electrical resistance (Ω/km)			Current carrying capacity A (I)
		Alum. (mm)	Steel (mm)	Total (mm²)	Alum. (mm²)	Total (mm)	Núcleo (mm)	Total Kg/Km	Alum. kg/km	Steel kg/km		d.c. 20°C	a.c. 25°C	a.c. 75°C	
TURKEY	6	6 x 1,68	1 x 1,68	15,52	13,30	5,04	1,68	53,8	36,5	17,3	5 295	2,1135	2,1496	2,6850	110
SWAN	4	6 x 2,12	1 x 2,12	24,71	21,18	6,36	2,12	85,4	58,0	27,4	8 280	1,3278	1,3537	1,7172	145
SWANATE	4	7 x 1,96	1 x 2,61	26,47	21,12	6,53	2,61	99,7	58,0	41,7	10 500	1,3133	1,3387	1,7383	145
SPARROW	2	6 x 2,67	1 x 2,67	39,20	33,60	8,01	2,67	135,9	92,3	43,6	12 680	0,8343	0,8527	1,1081	195
SPARATE	2	7 x 2,47	1 x 3,30	42,09	33,54	5,24	3,30	158,8	92,3	66,5	16 200	0,8251	0,8435	1,1181	195
ROBIN	1	6 x 3,00	1 x 3,00	49,48	42,41	9,00	3,00	171,4	116,4	55,0	15 800	0,6621	0,6768	0,8907	200
RAVEN	1/0	6 x 3,37	1 x 3,37	62,44	53,52	10,11	3,37	216,1	146,7	69,4	19 490	0,5243	0,5370	0,7165	255
QUAIL	2/0	6 x 3,78	1 x 3,78	78,55	67,33	11,34	3,78	272,5	185,0	87,5	23 630	0,4160	0,4265	0,5803	295
PIGEON	3/0	6 x 4,25	1 x 4,25	99,31	85,12	12,75	4,25	343,5	233,2	110,3	29 460	0,3304	0,3386	0,4705	340
PENGWIN	4/0	6 x 4,77	1 x 4,77	125,09	107,22	14,31	4,77	433,2	294,2	139,0	37 160	0,2618	0,2697	0,3829	390
WAXWING	266,8	18 x 3,09	1 x 3,09	142,5	135,0	15,45	3,09	431,6	372,9	58,7	30 620	0,2119	0,2169	0,2595	480
PARTRIDGE	266,8	26 x 2,57	7 x 2,00	156,9	134,9	16,28	6,00	546,1	374,3	171,8	50 280	0,2100	0,2146	0,2569	490



RESULTADOS MÉTODO DE BUSCA SEQUENCIAL - REDE DE 33 BARRAMENTOS

V.1 Cenário: Base

Iteração		Inicial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Linha	Ligada	--	18	16	15	13	11	10	30	33	12	11	13	15	16	24	37	14	20
	Desligada	--	16	15	13	11	10	8	33	32	11	13	15	16	--	37	36	20	19
Perdas (kW)		172,50	130,12	127,92	125,96	124,46	122,45	126,26	119,30	121,59	116,19	115,68	115,35	115,19	--	113,91	116,11	110,10	110,45
Tensão nodal (p.u) nos barramentos	1	1,0000					1,0000		1,0000					1,0000		1,0000		1,0000	
	2	0,9972					0,9972		0,9972					0,9972		0,9972		0,9972	
	3	0,9843					0,9876		0,9876					0,9876		0,9879		0,9879	
	4	0,9776					0,9834		0,9858					0,9858		0,9861		0,9861	
	5	0,9710					0,9793		0,9844					0,9844		0,9846		0,9846	
	6	0,9603					0,9733		0,9821					0,9821		0,9823		0,9823	
	7	0,9571					0,9727		0,9814					0,9814		0,9817		0,9817	
	8	0,9446					0,9586		0,9586					0,9739		0,9726		0,9710	
	9	0,9388					0,9603		0,9603					0,9725		0,9713		0,9669	
	10	0,9335					0,9626		0,9626					0,9716		0,9703		0,9660	
	11	0,9327					0,9630		0,9630					0,9715		0,9703		0,9659	
	12	0,9313					0,9640		0,9640					0,9687		0,9665		0,9703	
	13	0,9257					0,9584		0,9584					0,9631		0,9600		0,9678	
	14	0,9236					0,9563		0,9563					0,9611		0,9576		0,9671	
	15	0,9224					0,9550		0,9550					0,9598		0,9559		0,9653	
	16	0,9211					0,9538		0,9538					0,9585		0,9543		0,9636	
	17	0,9193					0,9519		0,9519					0,9567		0,9515		0,9609	
	18	0,9187					0,9514		0,9514					0,9561		0,9505		0,9599	
	19	0,9967					0,9954		0,9954					0,9954		0,9953		0,9953	
	20	0,9932					0,9802		0,9802					0,9802		0,9792		0,9792	
	21	0,9925					0,9760		0,9760					0,9760		0,9748		0,9748	
	22	0,9918					0,9691		0,9691					0,9717		0,9700		0,9722	
	23	0,9808					0,9841		0,9806					0,9806		0,9811		0,9811	
	24	0,9743					0,9776		0,9667					0,9667		0,9677		0,9677	
	25	0,9711					0,9744		0,9560					0,9560		0,9575		0,9575	
	26	0,9587					0,9717		0,9818					0,9818		0,9820		0,9820	
	27	0,9565					0,9695		0,9815					0,9815		0,9818		0,9818	
	28	0,9466					0,9596		0,9810					0,9810		0,9813		0,9813	
	29	0,9396					0,9526		0,9450					0,9450		0,9471		0,9471	
	30	0,9363					0,9493		0,9418					0,9418		0,9440		0,9440	
	31	0,9325					0,9455		0,9380					0,9380		0,9408		0,9408	
	32	0,9316					0,9447		0,9371					0,9371		0,9402		0,9402	
	33	0,9314					0,9444		0,9369					0,9369		0,9500		0,9594	

V.2 Cenário: Redução de carga

Iteração		Inicial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Linha	Ligada	--	18	16	15	13	11	10	30	33	12	11	13	15	16	24	37	14	20
	Desligada	--	16	15	13	11	10	8	33	32	11	13	15	16	--	37	36	20	19
Perdas (kW)		110,53	83,363	81,950	80,694	79,737	78,446	80,881	76,390	77,855	74,391	74,069	73,853	73,750	--	72,935	74,360	70,492	70,713
Tensão nodal (p.u.) nos barramentos	1	1,0000					1,0000		1,0000					1,0000		1,0000		1,0000	
	2	0,9978					0,9978		0,9978					0,9978		0,9978		0,9978	
	3	0,9874					0,9901		0,9901					0,9901		0,9903		0,9903	
	4	0,9821					0,9867		0,9887					0,9887		0,9889		0,9889	
	5	0,9768					0,9835		0,9875					0,9875		0,9877		0,9877	
	6	0,9683					0,9787		0,9856					0,9856		0,9858		0,9858	
	7	0,9657					0,9782		0,9852					0,9852		0,9853		0,9853	
	8	0,9557					0,9669		0,9669					0,9791		0,9781		0,9768	
	9	0,9511					0,9683		0,9683					0,9780		0,9770		0,9735	
	10	0,9468					0,9701		0,9701					0,9773		0,9763		0,9728	
	11	0,9462					0,9704		0,9704					0,9772		0,9762		0,9727	
	12	0,9451					0,9712		0,9712					0,9750		0,9732		0,9763	
	13	0,9406					0,9667		0,9667					0,9705		0,9680		0,9743	
	14	0,9389					0,9651		0,9651					0,9689		0,9661		0,9737	
	15	0,9379					0,9640		0,9640					0,9678		0,9647		0,9722	
	16	0,9369					0,9630		0,9630					0,9668		0,9634		0,9709	
	17	0,9354					0,9616		0,9616					0,9653		0,9612		0,9687	
	18	0,9350					0,9611		0,9611					0,9649		0,9604		0,9679	
	19	0,9974					0,9963		0,9963					0,9963		0,9963		0,9963	
	20	0,9945					0,9842		0,9842					0,9842		0,9833		0,9833	
	21	0,9940					0,9808		0,9808					0,9808		0,9798		0,9798	
	22	0,9935					0,9753		0,9753					0,9774		0,9760		0,9777	
	23	0,9846					0,9873		0,9845					0,9845		0,9849		0,9849	
	24	0,9795					0,9821		0,9733					0,9733		0,9741		0,9741	
	25	0,9769					0,9795		0,9648					0,9648		0,9660		0,9660	
	26	0,9669					0,9774		0,9854					0,9854		0,9856		0,9856	
	27	0,9652					0,9756		0,9852					0,9852		0,9854		0,9854	
	28	0,9573					0,9677		0,9848					0,9848		0,9850		0,9850	
	29	0,9516					0,9621		0,9560					0,9560		0,9576		0,9576	
	30	0,9490					0,9595		0,9534					0,9534		0,9552		0,9552	
	31	0,9460					0,9564		0,9504					0,9504		0,9527		0,9527	
	32	0,9453					0,9557		0,9497					0,9497		0,9522		0,9522	
	33	0,9451					0,9555		0,9495					0,9495		0,9600		0,9675	

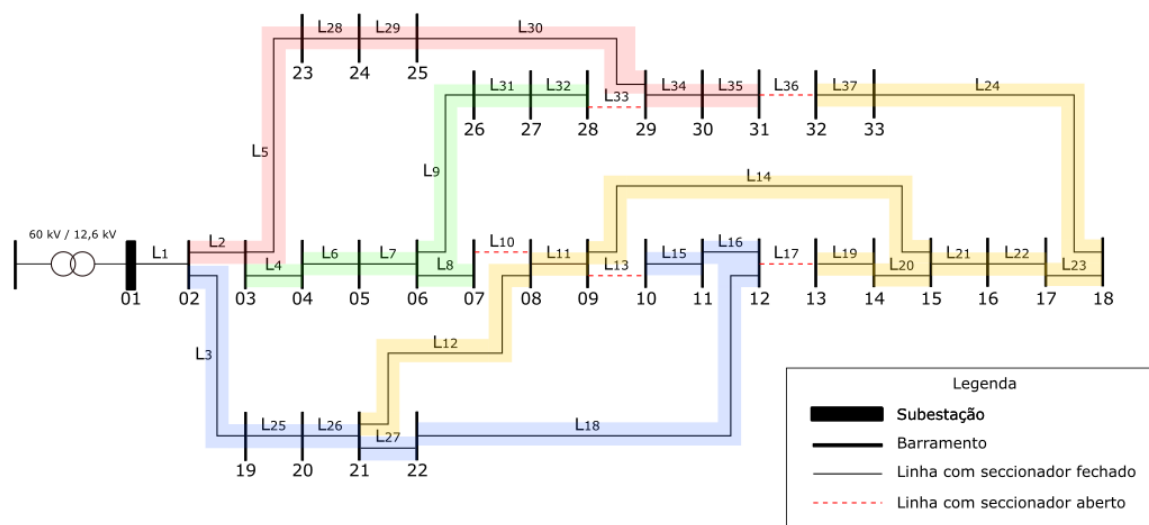
V.3 Cenário: Aumento de carga

Iteração		Inicial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Linha	Ligada	--	18	16	15	13	11	10	30	33	12	11	13	15	16	24	37	14	20
	Desligada	--	16	15	13	11	10	8	33	32	11	13	15	16	--	37	36	20	19
Perdas (kW)		248,10	187,18	184,02	181,20	179,05	176,16	181,63	171,72	175,02	167,24	166,52	166,04	165,80	--	163,96	167,08	158,49	158,99
Tensão nodal (p.u.) nos barramentos	1	1,0000					1,0000		1,0000					1,0000		1,0000		1,0000	
	2	0,9967					0,9967		0,9967					0,9967		0,9967		0,9967	
	3	0,9811					0,9851		0,9851					0,9851		0,9854		0,9854	
	4	0,9731					0,9800		0,9830					0,9830		0,9833		0,9833	
	5	0,9651					0,9752		0,9813					0,9813		0,9816		0,9816	
	6	0,9524					0,9680		0,9785					0,9785		0,9788		0,9788	
	7	0,9485					0,9673		0,9777					0,9777		0,9780		0,9780	
	8	0,9335					0,9503		0,9503					0,9687		0,9672		0,9652	
	9	0,9266					0,9524		0,9524					0,9670		0,9655		0,9603	
	10	0,9202					0,9551		0,9551					0,9659		0,9644		0,9592	
	11	0,9192					0,9556		0,9556					0,9659		0,9643		0,9591	
	12	0,9176					0,9568		0,9568					0,9625		0,9598		0,9644	
	13	0,9109					0,9501		0,9501					0,9558		0,9520		0,9614	
	14	0,9084					0,9476		0,9476					0,9533		0,9491		0,9605	
	15	0,9068					0,9461		0,9461					0,9517		0,9471		0,9583	
	16	0,9053					0,9446		0,9446					0,9502		0,9451		0,9563	
	17	0,9031					0,9423		0,9423					0,9480		0,9418		0,9530	
	18	0,9024					0,9417		0,9417					0,9473		0,9406		0,9519	
	19	0,9961					0,9945		0,9945					0,9945		0,9944		0,9944	
	20	0,9918					0,9762		0,9762					0,9762		0,9750		0,9750	
	21	0,9910					0,9712		0,9712					0,9712		0,9697		0,9697	
	22	0,9902					0,9630		0,9630					0,9661		0,9640		0,9666	
	23	0,9769					0,9809		0,9768					0,9768		0,9773		0,9773	
	24	0,9692					0,9732		0,9600					0,9600		0,9612		0,9612	
	25	0,9653					0,9693		0,9472					0,9472		0,9490		0,9490	
	26	0,9504					0,9660		0,9781					0,9781		0,9784		0,9784	
	27	0,9478					0,9634		0,9778					0,9778		0,9781		0,9781	
	28	0,9359					0,9516		0,9772					0,9772		0,9775		0,9775	
	29	0,9275					0,9431		0,9341					0,9341		0,9365		0,9365	
	30	0,9235					0,9392		0,9301					0,9301		0,9329		0,9329	
	31	0,9190					0,9346		0,9255					0,9255		0,9290		0,9290	
	32	0,9180					0,9336		0,9245					0,9245		0,9282		0,9282	
	33	0,9176					0,9333		0,9242					0,9242		0,9400		0,9513	

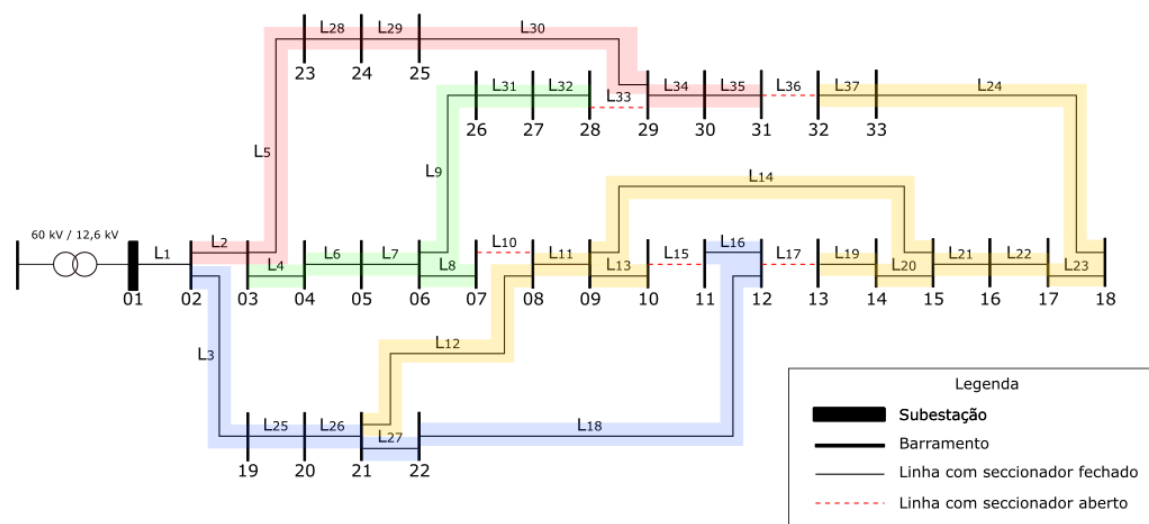
RESULTADOS ALGORITMO GENÉTICO - REDE DE 33 BARRAMENTOS

VI.1 Cenário: Base

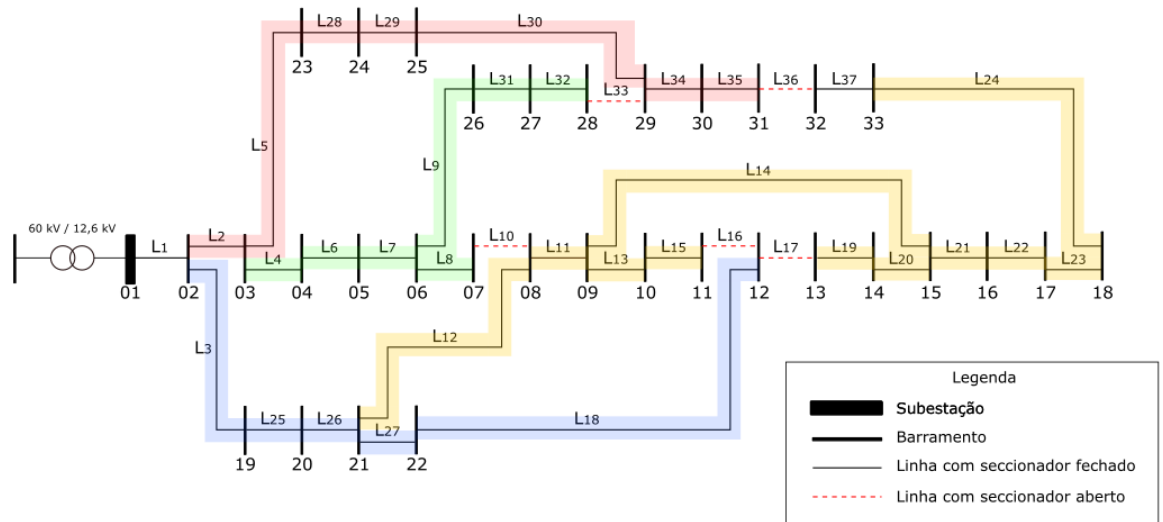
1. Rede nº 4



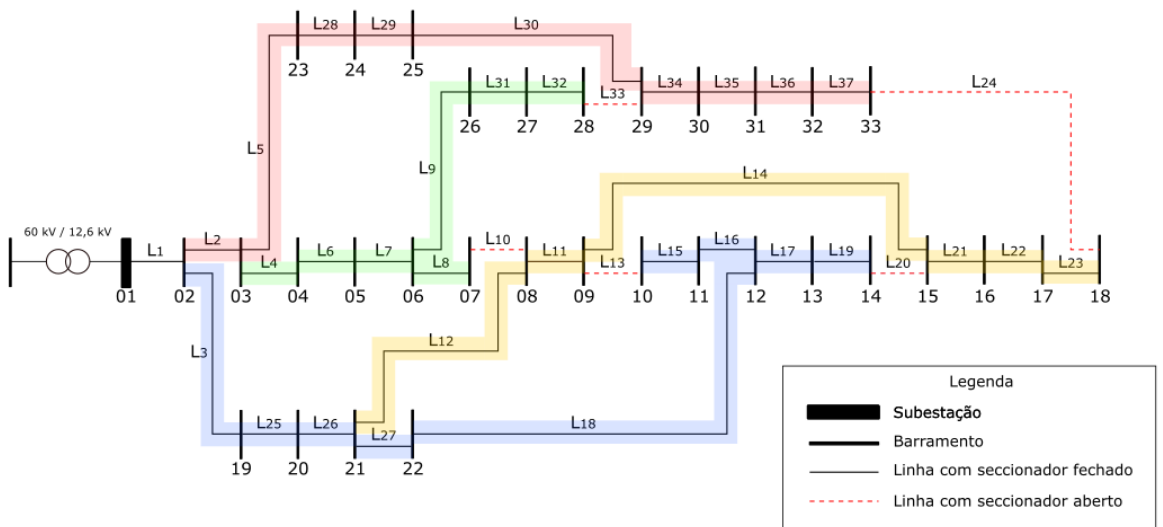
2. Rede nº 5



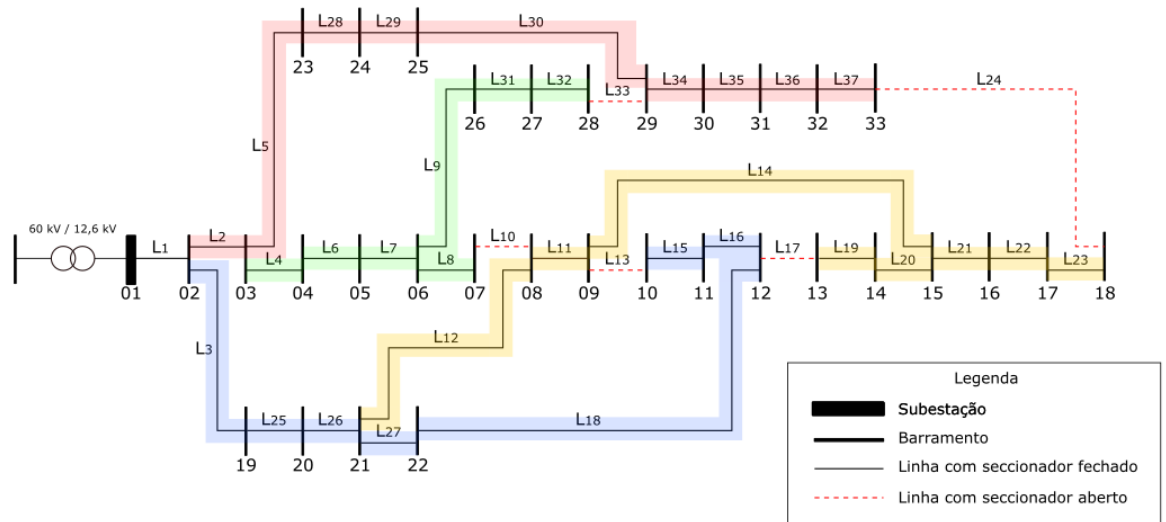
3. Rede nº 6



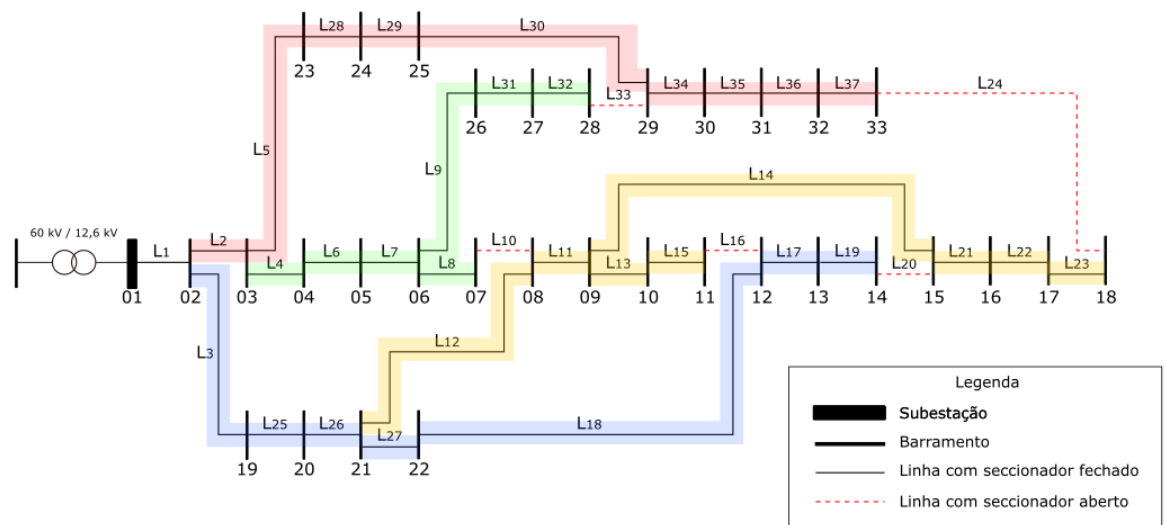
4. Rede nº 7



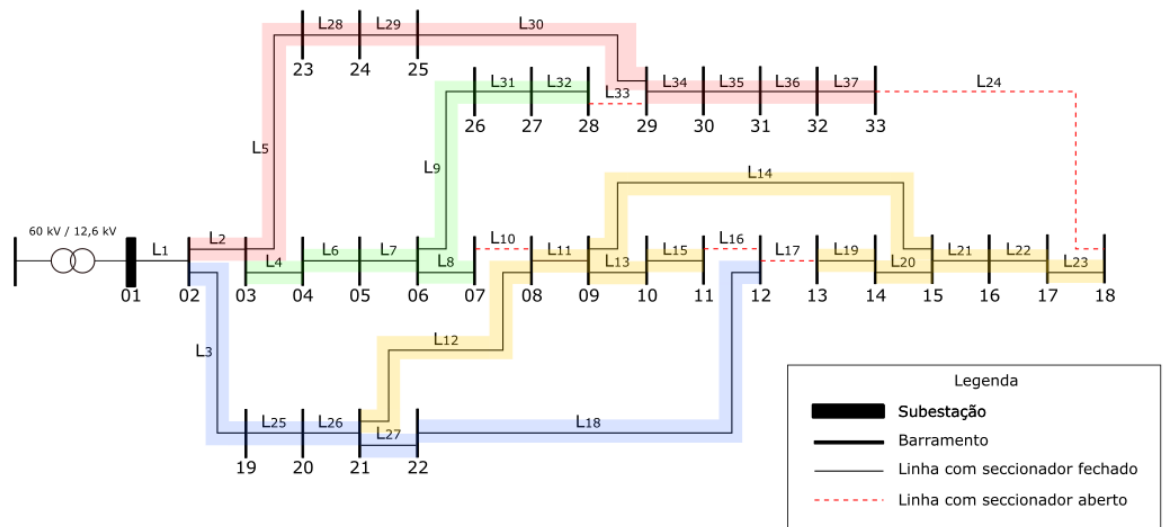
5. Rede nº 8



6. Rede nº 9

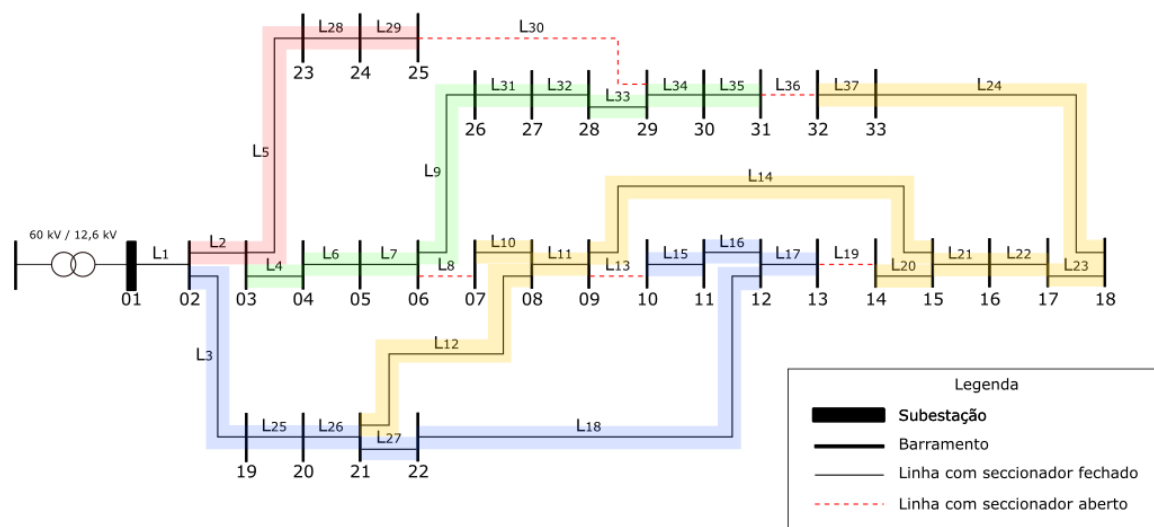


7. Rede nº 10

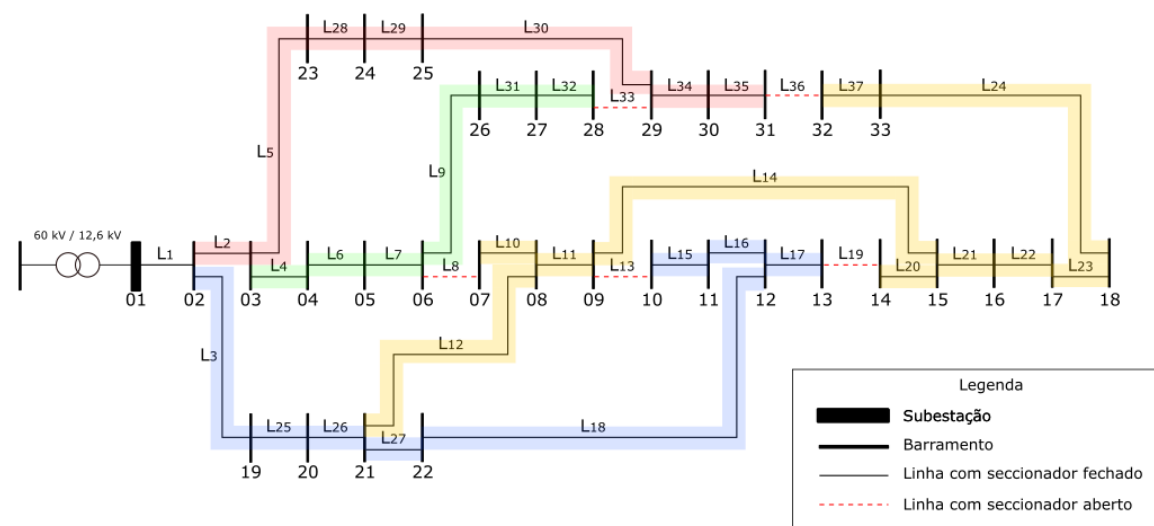


VI.2 Cenário: Redução de carga

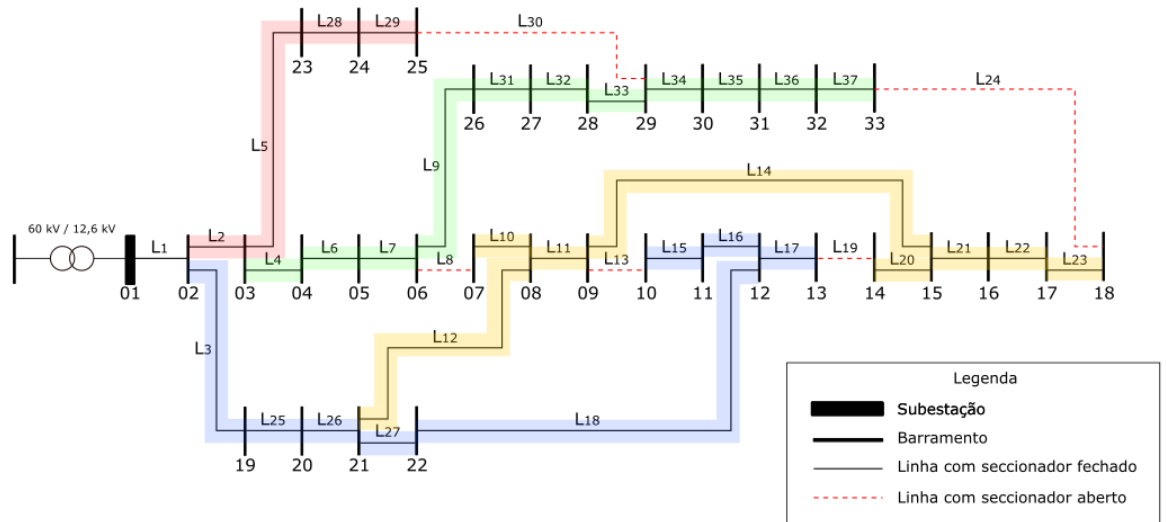
1. Rede nº 4



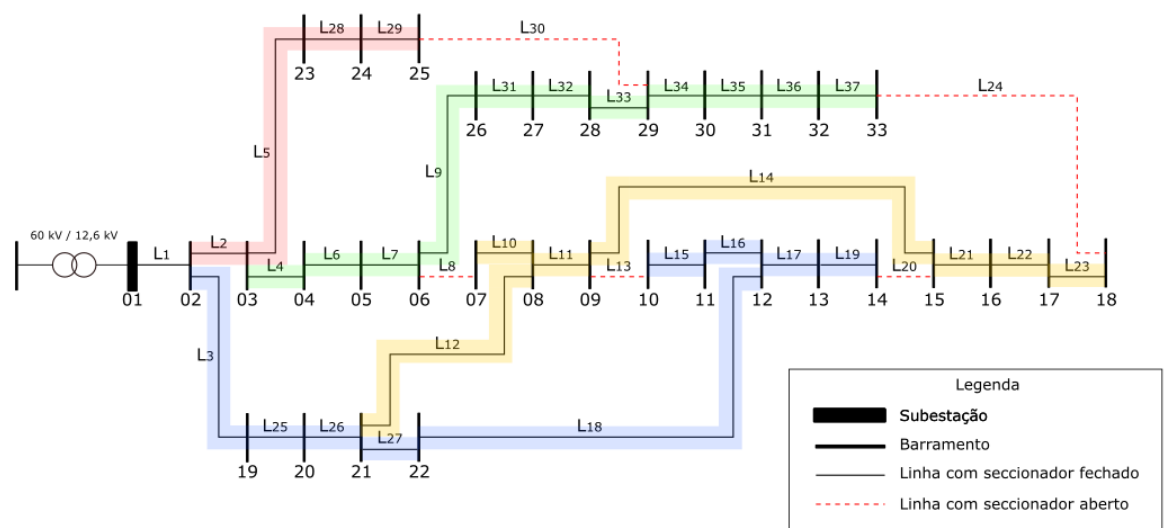
2. Rede nº 5



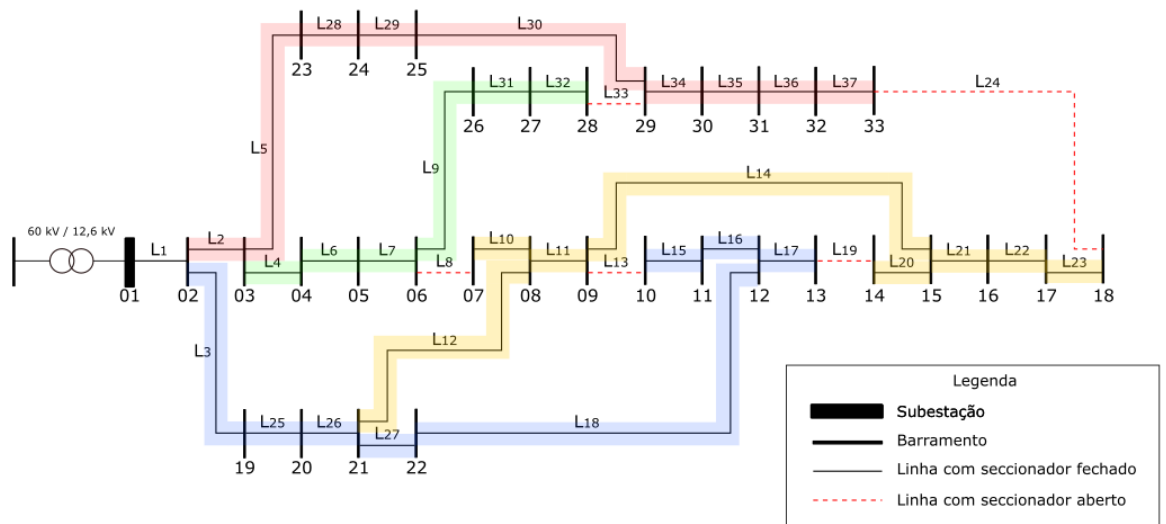
3. Rede nº 6



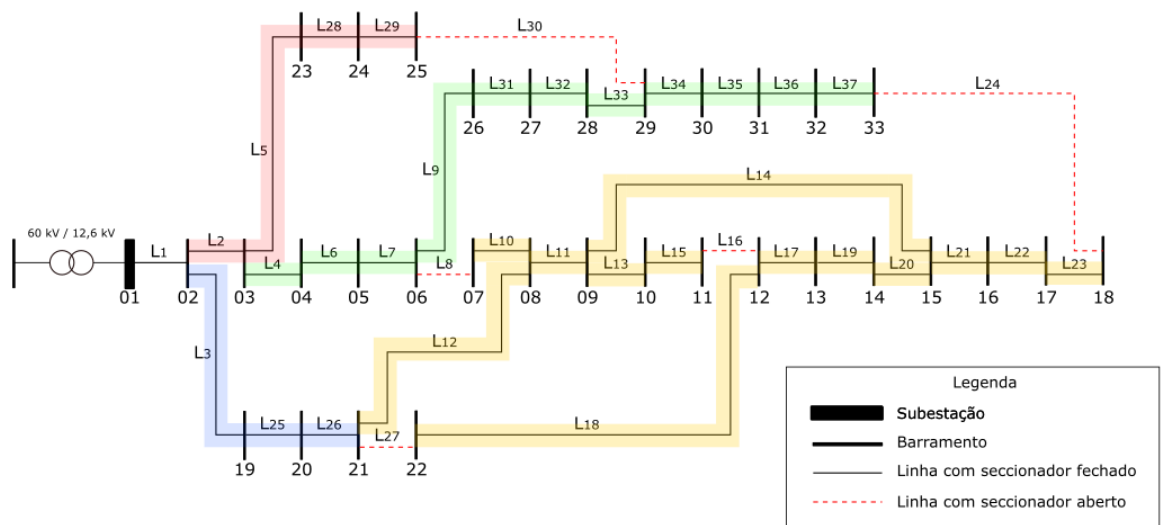
4. Rede nº 7



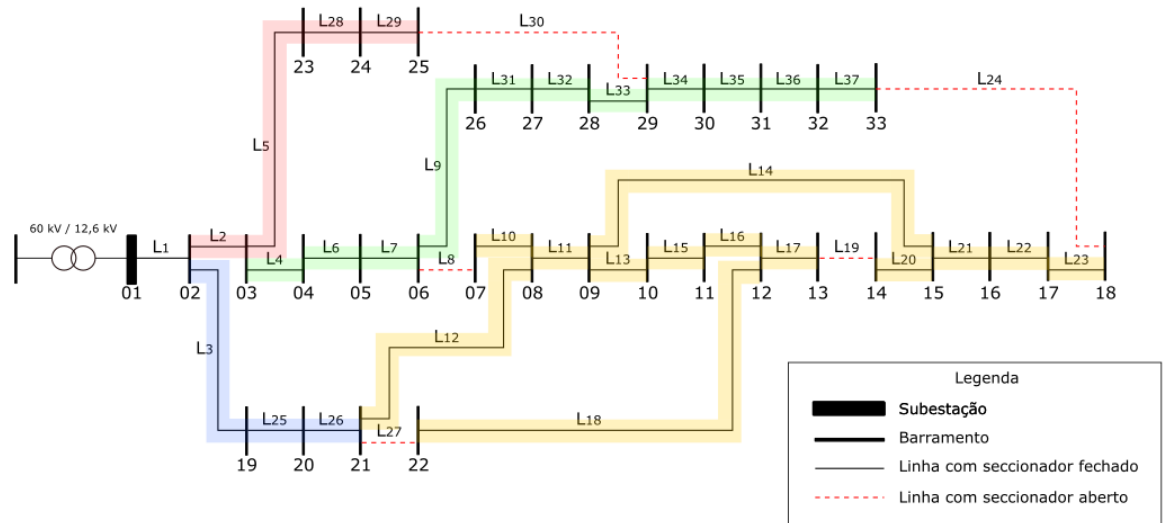
5. Rede nº 8



6. Rede nº 9

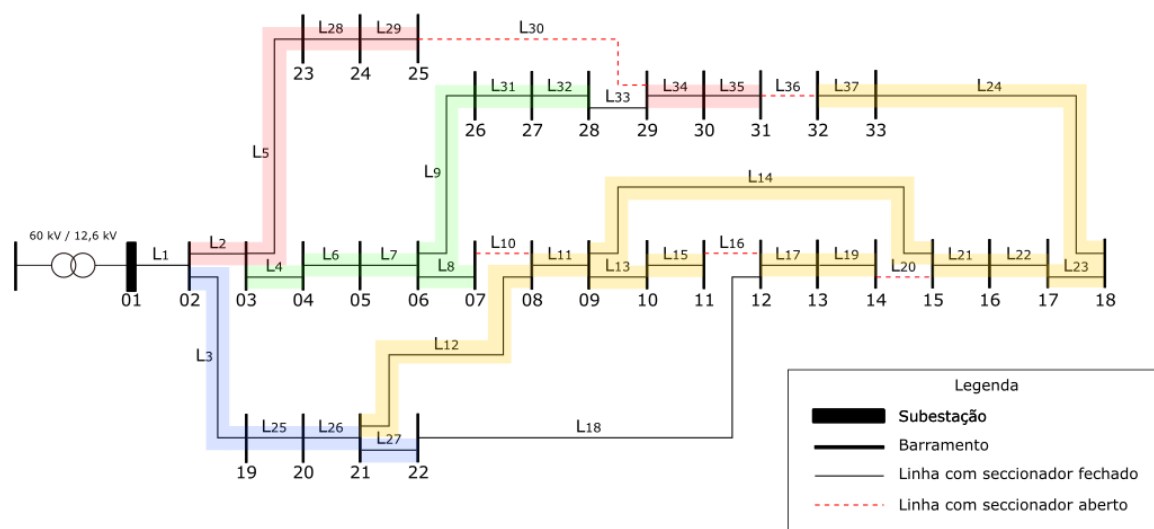


7. Rede nº 10

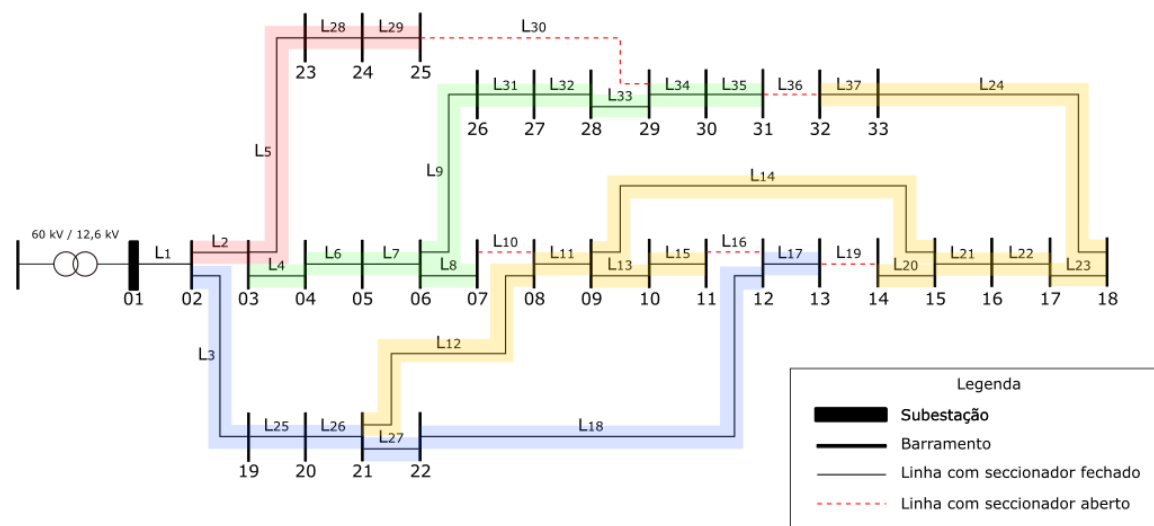


VI.3 Cenário: Aumento de carga

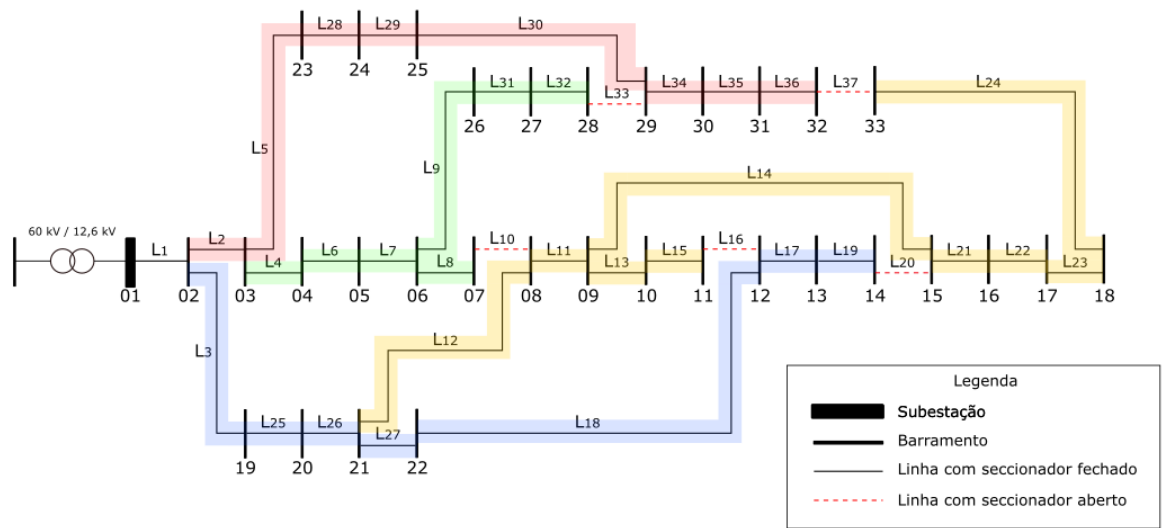
1. Rede nº 4



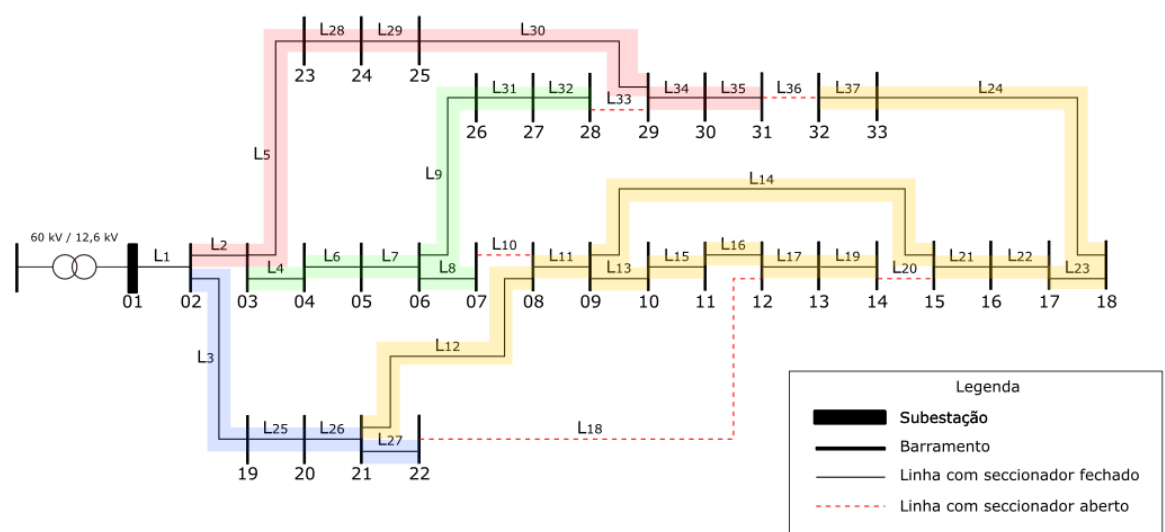
2. Rede nº 5



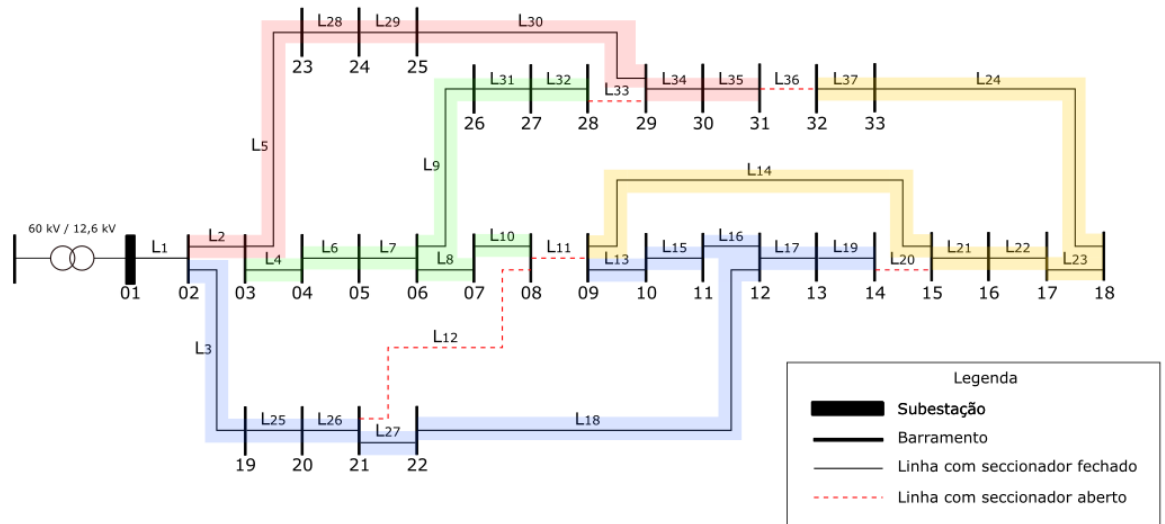
3. Rede nº 6



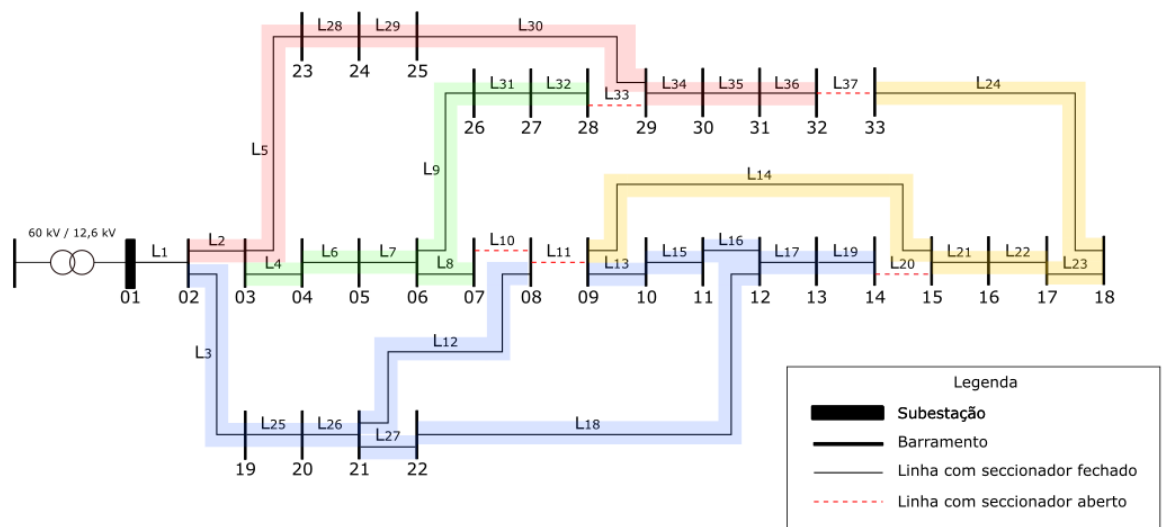
4. Rede nº 7



5. Rede nº 8



6. Rede nº 9



7. Rede nº 10

